Uncertainty and the Medical Interview
Towards Self-Assessment in Machine Learning Models

Havtorn, Jakob Drachmann

[Link back to DTU Orbit](#)

UNCERTAINTY AND THE MEDICAL INTERVIEW

TOWARDS SELF-ASSESSMENT IN MACHINE LEARNING MODELS

Jakob Drachmann Havtorn

January 2024

Copenhagen

ACADEMIC SUPERVISION
Technical University of Denmark

Jes Frellsen
Supervisor

Søren Hauberg
Co-supervisor

Ole Winther
Co-supervisor

INDUSTRIAL SUPERVISION
Corti

Lars Maaløe
Supervisor

Zeljko Agić
Co-supervisor

DECLARATION
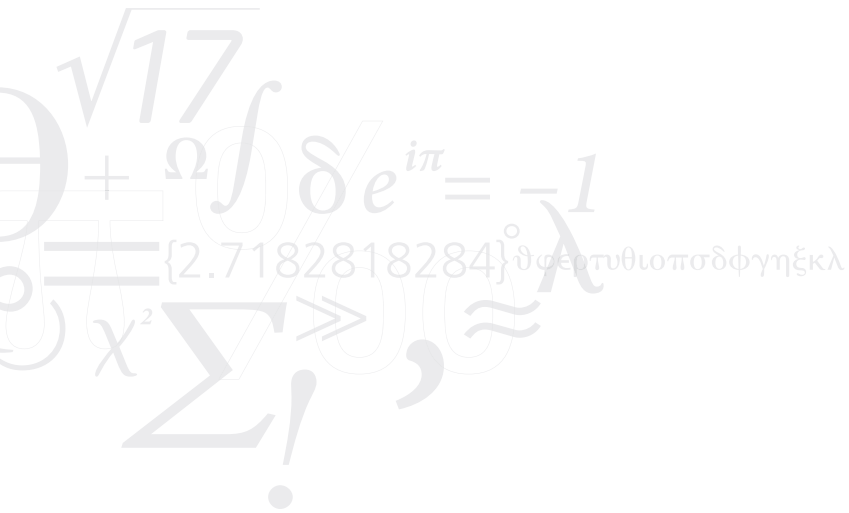
This PhD thesis was prepared at the Department of Applied Mathematics and Computer Science at the Technical University of Denmark in fulfillment of the requirements for acquiring a PhD degree.

Copenhagen, January 1, 2024

Jakob Drachmann Havtorn

ABSTRACT

Natural language plays a key role in healthcare systems worldwide; yet, the medical interview process has seen little development compared to the strides made in medical technology over the past decade. Traditionally, medical interviews are conducted by healthcare professionals who primarily rely on their individual experience to understand a situation. Faced with aging populations, rigorous documentation requirements, and advances in diagnostic capabilities and treatment options, this approach is costly and risks falling short, potentially compromising the accuracy and quality of medical care.

Recent advances in natural language processing enable machine learning models to actively engage in medical interviews to alleviate administrative burdens, enhance documentation, and provide real-time assistance in dialogue. However, healthcare stands apart from other domains due to the high risk associated with even minor errors. Since no model is error-free, this impels pairing model predictions with robust uncertainty estimates, especially in scenarios involving out-of-distribution (OOD) data, such as rare illnesses.

This thesis sets out from the main hypothesis that unsupervised representation learning is useful for uncertainty estimation for medical tasks. It makes the following contributions:

(a) A likelihood-ratio score for OOD detection with variational autoencoders that alleviates the proved impeding effect of low-level features.

(b) A statistical test for OOD detection combining the score and typicality tests and applicable with likelihoods from any differentiable generative model.

(c) A benchmark of probabilistic speech representation learners and a novel method to learn hierarchical representations.

(d) An overview of unsupervised representation learning for neural speech processing and a corresponding model taxonomy.

(e) An error analysis and revised evaluation of state-of-the-art models for automated medical coding on the MIMIC-III and IV datasets.

(f) A retrospective study of speech-based stroke recognition in prehospital medical helpline calls with significant improvements over call-taker performance.

In summary, this thesis addresses challenges in uncertainty estimation and representation learning for speech while exploring medical applications of machine learning. Its contributions are vital in the development of an operational decision support system for medical interviews, ultimately aiming to improve the quality of patient care by supporting effective, informed decision-making.

RESUMÉ (ABSTRACT IN DANISH)

Naturligt sprog spiller en nøglerolle i sundhedssystemer verden over. Alligevel har den medicinske interviewproces kun oplevet en lille udvikling sammenlignet med fremskridtene inden for medicinsk teknologi. Traditionelt udføres medicinske interviews af sundhedspersonale, der primært afhænger af deres individuelle erfaring for at forstå en situation. Med aldrende befolkninger, strenge dokumentationskrav, og fremskridt inden for diagnostiske muligheder og behandlingsmuligheder, er denne tilgang dyr og risikerer at komme til kort, hvilket potentielt kompromitterer nøjagtigheden og kvaliteten af medicinsk behandling.

Nylige fremskridt inden for naturlig sprogbehandling gør det muligt for maskinlæringsmodeller at deltage aktivt i medicinske interviews for at lette administrative byrder, forbedre dokumentation, og assistere i realtid. Sundhedsplejen adskiller sig dog fra andre domæner på grund af den høje risiko forbundet med selv små fejl. Da ingen model er fejlfri, tilskynder dette til at associere modelprædiktioner med robuste usikkerhedsestimater, især i scenarier, der involverer ude-af-fordeling (UAF) data, såsom sjældne sygdomme.

Denne afhandling tager udgangspunkt i hovedhypotesen, at usuperviseret repræsentationslæring er nyttig til usikkerhedsestimering i medicinske opgaver. Den giver følgende bidrag:

(a) En likelihood-ratio score til UAF-detektion med variationelle autoenkodere, der afhjælper den bevist negative effekt af lav-niveau features.

(b) En statistisk test til UAF-detektion, der kombinerer score- og typikalitetstests og kan bruges med likelihoods fra enhver differentiabel generativ model.

(c) Et benchmark for probabilistiske talerepræsentationsmodeller og en ny metode til at lære hierarkiske repræsentationer.

(d) En oversigt over usuperviseret repræsentationslæring til neural talebehandling og en tilsvarende modeltaksonomi.

(e) En fejlanalyse og revideret evaluering af state-of-the-art modeller til automatiseret medicinsk kodning på MIMIC-III og IV datasættene.

(f) Et retrospektivt studie af talebaseret genkendelse af stroke i præhospitale akuttelefonopkald der viser betydelig forbedring i forhold til opkaldstagere.

Sammenfattende adresserer denne afhandling udfordringer indenfor usikkerhedsestimering og repræsentationslæring til tale og udforsker medicinske anvendelser af maskinlæring. Dens bidrag er afgørende i udviklingen af et operationelt beslutningsstøttesystem til medicinske interviews, der søger at øge kvaliteten af patientbehandlingen ved at understøtte effektiv, informeret beslutningstagen.

Publications

The present thesis comprises the following set of research papers based on the candidate's original research.

PRIMARY

[A] **Havtorn, J. D.**, Frellsen, J., Hauberg, S., Maaløe, L., "Hierarchical VAEs Know What They Don't Know". In: *Proceedings of the 38th International Conference on Machine Learning (ICML)*. Virtual: PMLR, 2021. arXiv: 2102.08248 [main author]

[B] Bergamin, F., Mattei, P.-A., **Havtorn, J. D.**, Senetaire, H., Schmutz, H., Maaløe, L., Hauberg, S., Frellsen, J., "Model-Agnostic Out-of-Distribution Detection Using Combined Statistical Tests". In: *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*. volume 151. Valencia, Span: PMLR, 2022. arXiv: 2203.01097 [coauthor]

[C] Borgholt, L., **Havtorn, J. D.**, Edin, J., Maaløe, L., Igel, C., "A Brief Overview of Neural Speech Representation Learning". In: *Proceedings of the 2nd Workshop on Self-supervised Learning for Audio and Speech Processing (SAS) at the Thirty-Sixth AAAI Conference on Artificial Intelligence*. Virtual, 2022. arXiv: 2203.01829 [coauthor]

[D] **Havtorn, J. D.**, Borgholt, L., Hauberg, S., Frellsen, J., Maaløe, L., "Benchmarking Generative Latent Variable Models for Speech". In: *Proceedings of the Workshop on Deep Generative Models for Highly Structured Data at ICML*. 2022. arXiv: 2202.12707 [main author]

[E] Edin, J., Junge, A., **Havtorn, J. D.**, Borgholt, L., Maistro, M., Ruotsalo, T., Maaløe, L., "Automated Medical Coding on MIMIC-III and MIMIC-IV: A Critical Review and Replicability Study". In: *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. Taipei, Taiwan: ACM, 2023. arXiv: 2304.10909 [coauthor]

[F] Wenstrup, J., **Havtorn, J. D.**, Borgholt, L., Blomberg, S. N., Maaloe, L., Sayre, M. R., Christensen, H., Kruuse, C., Christensen, H. C., "A Retrospective Study on Machine Learning-Assisted Stroke Recognition for Medical Helpline Calls". In: *npj Digital Medicine* 6.1 (2023). ISSN: 2398-6352 [shared main author]

SECONDARY

[G] Mohamed, A., Lee, H.-y., Borgholt, L., **Havtorn, J. D.**, Edin, J., Igel, C., Kirchhoff, K., Li, S.-W., Livescu, K., Maaløe, L., Sainath, T. N., Watanabe, S., "Self-Supervised Speech Representation Learning: A Review". In: *IEEE Journal of Selected Topics in Signal Processing (JSTSP)* 16.6 (2022). arXiv: 2205.10643 [shared main author]

[H] Borgholt, L., Tax, T. M. S., **Havtorn, J. D.**, Maaløe, L., Igel, C., "On Scaling Contrastive Representations for Low-Resource Speech Recognition". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Virtual: IEEE, 2021. arXiv: 2102.00850 [coauthor]

[I] Borgholt, L., **Havtorn, J. D.**, Abdou, M., Edin, J., Maaløe, L., Søgaard, A., Igel, C., *Do We Still Need Automatic Speech Recognition for Spoken Language Understanding?* 2021. arXiv: 2111.14842 [coauthor]

Papers labeled as PRIMARY are central to the main theme of the project and are included as chapters of the thesis. Although published as part of the project, the SECONDARY papers are less central to the main theme and are not included in the thesis. Paper [G] is included in the appendix as an extended version of [C] with a focus on self-supervised methods.

The included papers have had their layout and formatting adjusted to conform to that of the thesis and any spelling errors and minor language errors have been corrected. The scientific content has remained unchanged.

Acknowledgements

First, I want to thank my supervisors. Jes, you have been an amazing support through this project. Your positivity and insightfulness were essential to me daring to take on this project, and to its success. Lars, you have been a completely indispensable source of purpose, energy, and direction throughout my time with Corti. I look forward to many more fruitful discussions in the future.

I also want to thank the brilliant researchers at the section for Cognitive Systems that I have had the privilege of working with. Søren, as my co-supervisor, you have been a fantastic discussion partner. Your wit and your ability to pose genuinely significant questions have been invaluable. I want to also convey appreciation for the researchers in my group, for the open, inclusive, and relaxed culture, and for many good discussions, often over coffee and pastries.

I want to thank Lars, Andreas, and Michael for letting me become part of the Corti project after graduating back in 2018. The foundational will to invest and believe in people, and to dare to have bold visions, has made Corti a fantastic place for me to develop, professionally and personally. I want to also thank the members of the machine learning team during the first years of my time at Corti. Lasse, Tycho, Marco, Jan, Janek, and Joakim, you all played a part in inspiring and enabling me to dive into this project. A special thanks should go to Joakim. Your honest curiosity and dedication has been a wonderful addition to the research team. I also want to thank Lasse. The numerous inspirational discussions with you and our combined efforts through the years have been instrumental in shaping this thesis. It has been a privilege to carry out this PhD project in the company of you both.

I want to also thank my friends and my family who have provided much needed diversion and encouragement throughout the years. Last, but not least, I want to thank my wife, Rikke, who has been an unbelievably patient and absolutely essential supporter at every step of this undertaking. I am excited about what lies ahead for us and our daughter Vigga, and I cannot wait to build our future together.

Contents

PART I

BACKGROUND

CHAPTER 1

INTRODUCTION

In an era where technology increasingly intertwines with healthcare, artificial intelligence is on the verge of becoming part of standard medical practice. Growing administrative burdens, aging populations, and rapid scientific progress in medicine and medical devices have created a need to rethink how medical professionals are best enabled to successfully do their job. The application of artificial intelligence in medical decision-making is likely to hold significant potential to reduce the effect of these issues and improve patient outcomes [595]. However, as such decisions impact critical aspects of human health and well-being, this prospect has also raised concerns relating to the reliability of systems that use artificial intelligence [1, 94].

The use of technology to support decision-making can be traced back to very beginning of recorded history. After the agricultural revolution, several ancient civilizations developed mathematical techniques and algorithms to help them manage the newfound complexity of society. Fittingly, the first human name recorded in written history belonged to Kushim, a Sumerian accountant. Their signature has been identified on at least 18 clay tablets dated to 3400-3000 BCE, documenting transactions and inventory related to barley and other ingredients used in beer production [490]. Later, the Babylonians developed a sophisticated system of mathematics that included algorithms for solving quadratic equations and calculating areas of shapes, and numerical methods for approximating square roots [187].

Besides enabling the advancement of early society, these technologies also brought with them novel challenges. In a likely attempt to ease the strenuous work of doing Sumerian mathematics, Kushim produced a clay tablet with a multiplication table that included fractional representations of common numbers (figure 1.1). Assumedly unbeknownst to Kushim, this tablet would also come to contain the world's first recorded example of a mathematical error; the statement: $5 \times 1/2 = 5$ [490]. Later, the scribe of the Plimpton 322 clay tablet also made several errors when listing Pythagorean triplets, presumably struggling to grapple with Babylonian mathematics [63, 139, 481]. Besides these human errors, limitations inherent to the tools themselves also posed problems. The Babylonian number system, for instance, was well-equipped to represent rational numbers but inefficient at approximating irrational numbers. This forced significant approximation errors such as the common practice of approximating π with 3 [203].

While efforts to ease human decision-making have been ongoing for thou-

Figure 1.1: Technology is difficult. Between 3400 and 3000 BCE, the Sumerian accountant Kushim wrote this tablet. Besides calculations of basic ingredients required for the production of cereal products, in the top left 6-by-2 grid it contains a multiplication table. While presumably aimed to ease the strenuous work of doing Sumerian mathematics, the top row unfortunately states: $5 \times {}^1/_2 = 5$, an unforgiving imprint of the world's first recorded mathematical error [490, 554], [photo credit 140].

sands of years, it was not until 1642 that the first mechanical calculator was successfully designed by Blaise Pascal. Refined by Gottfried Leibniz during the latter half of the 17th century, the mechanical calculator marked the beginning of a new era where rudimentary calculations could be automated and performed with greater speed and accuracy than by humans alone. In the 1820s, Charles Babbage proposed the first programmable mechanical computer, the Difference Engine. Although never completed, it laid the foundation for his later design of the Analytical Engine in 1837 which is considered to be the first general-purpose computer design. Working with Babbage, Ada Lovelace is widely regarded as
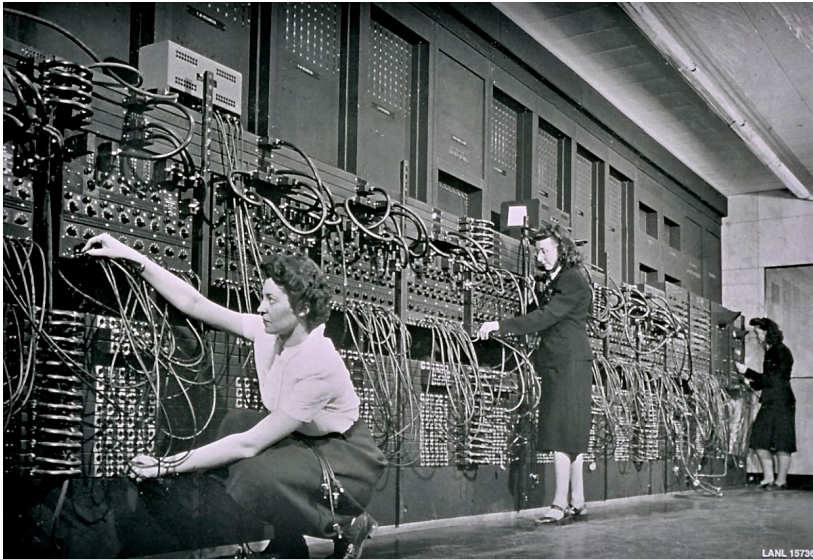
Figure 1.2: Ruth Lichterman (left) and Marlyn Wescoff (middle) were two of the several female programmers of the ENIAC. [photo credit 656]

the first to have recognized that programmable computers could have applications beyond pure arithmetic. However, only with the advent of electronics were the first successful general-purpose computers built, starting with the ENIAC in 1945 (figure 1.2). As the technology matured and manufacturing processes were refined and scaled, electronic computers were adopted widely across society for various applications in science and industry, including healthcare [203, 235].

With the digital revolution continuing into the 21st century and the worldwide spread of the internet, the amount of data and its availability have grown exponentially. This has led to a surge of interest in the scientific field of machine learning, a subfield of artificial intelligence, that studies how computer algorithms can learn functions from data and make predictions to guide decision-making. Enabled by a massive increase in the performance and availability of parallel computing resources, such methods have set new standards for the abilities of computer systems at various tasks usually reserved for humans, such as vision, speech recognition, and natural language understanding [377]. In recent years, machine learning has also made its way into healthcare systems, where it has shown promising results on difficult tasks such as medical imaging [428], drug discovery [93], and clinical decision support [52, 53].

Machine learning, like previous technologies, carries both benefits and risks. However, machine learning might exacerbate some of these risks due to the
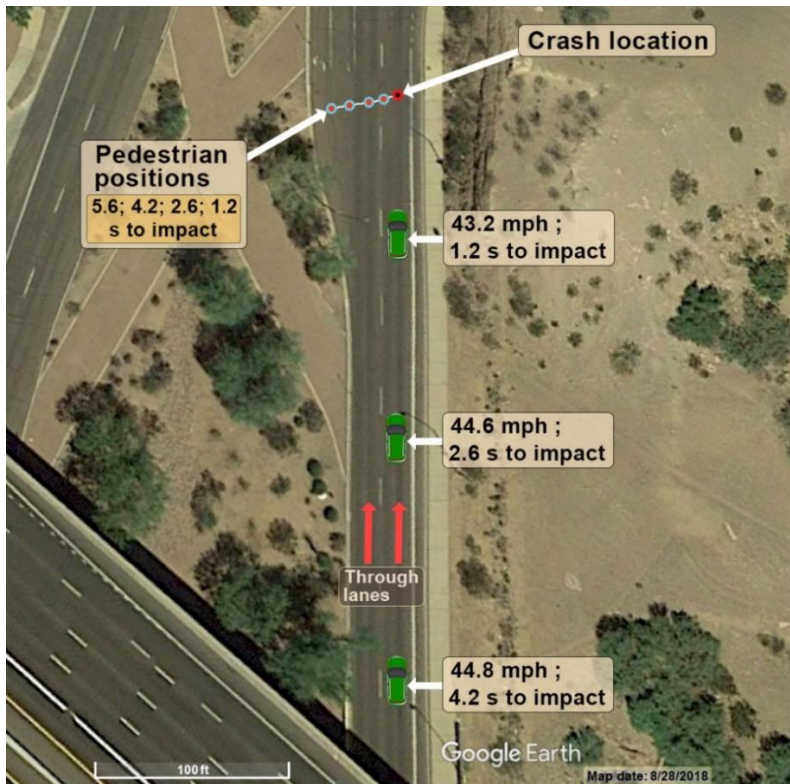
Figure 1.3: A pedestrian was killed by an Uber self-driving car in Tempe, Arizona in 2018. The car's sensors detected the pedestrian 5.6 seconds before the crash, but the self-driving system failed to recognize its own uncertainty in multiple object misclassifications and so did not correctly predict her path or reduce the SUV's speed [475, photo credit].

unique tasks it enables in domains such as healthcare and autonomous driving. While applications in healthcare are nascent and generally move forward in small steps, autonomous driving has fast adopted machine learning and presents a pertinent example: In 2018, the first incident of a self-driving car killing a pedestrian took place in Tempe, Arizona. The car, an SUV operated by Uber, was driving in autonomous mode when it struck a pedestrian crossing the street with her bicycle. According to the investigation by the National Transportation Safety Board, the car's sensors detected the pedestrian more than five seconds before the crash, but the system did not manage to correctly predict her path or reduce the SUV's speed. Specifically, during those seconds, the system incorrectly classified the

pedestrian more than ten times, first as a vehicle, then as an unknown object and ultimately as a bicyclist, each time changing, or resetting, the pedestrian's predicted path. About one second before the crash, the system determined that a collision was imminent, but the situation exceeded the constraints within which the autonomous driving system was allowed to operate, and the car's safety driver failed to intervene (figure 1.3) [475]. And Uber is not the only actor facing the challenges of applying machine learning in real-world scenarios. A recent report by the Washington Post found that Tesla's Autopilot system has been activated during at least 736 crashes in the last four years with 16 fatalities [605]. While driver-assistance systems undoubtedly help reduce accidents in many cases, and humans have been argued to be worse drivers still, the issues of current systems highlight the need for machine learning systems that are robust in rare, adverse, and difficult situations [452].

Recently, European policymakers proposed legislation to enforce that applied machine learning meets certain standards [177]. This so-called AI Act considers three categories of applications based on the potential risk they pose to society: limited, high, and unacceptable. Autonomous vehicles and systems within healthcare fall in the category of high risk applications. To avoid critical errors, these must meet a set of strict requirements focusing on antidiscrimination and robustness.

One of the key advances needed to ensure that high-risk applications of machine learning comply with such requirements is accurate uncertainty estimation of machine predictions. The multitude of rapid misclassifications made by the Uber self-driving car in Tempe is a likely indication that the model responsible for object-classification was overconfident in its predictions and unable to represent or communicate this uncertainty to other subsystems in the car, or the safety driver. As a consequence, the model's many incoherent predictions were taken at face value and other subsystems acted on them without appropriate consideration to their reliability. While any model will inevitably make wrong predictions, accurately estimating the associated uncertainty and communicating it to reliant systems or humans is necessary to reduce the risk that wrong predictions lead to catastrophic failures. Fittingly, a recent study by the European Parliamentary Research Services [178] concluded:

> *Future AI solutions for healthcare should be implemented by integrating uncertainty estimation, a relatively new field of research that aims to provide clinicians with clinically useful indications on the degree of confidence in AI predictions.*

———————

The research presented in this thesis deals with the use of machine learning for medical interviews with a focus on uncertainty estimation and representation learning. The research project was partially funded by a grant from the national Danish Innovation Fund (grant no. 0153-00167B) and defined and carried out in collaboration with the Danish health tech company Corti, which develops computer software for medical decision support. In the rest of this chapter, we will provide a high-level introduction to these topics, and to Corti, and discuss the motivation behind the research project.

1.1 Uncertainty by example: Corti use-cases

Corti develops decision support software for healthcare professionals such as general practitioners, medical coders, and call-takers at emergency medical services. The system supports the professional similar to a copilot for medical interviews, for instance providing a graph-based protocol for triaging and helping to keep track of key information shared in the conversation.

When used by emergency services for example, calls are transcribed in realtime using automatic speech recognition built for the customer-domain and the system makes suggestions for the call-taker to use in the conversation with the caller, including notifications about forgotten protocol questions [246] and potential cases of cardiac arrests [52, 53]. The system logs the details about the call and the actions taken by the call-taker for later use in review, training and quality assurance.

1.1.1 Stroke recognition in emergency calls

Case background In 2021, Corti entered a research collaboration with the Capital Region of Denmark and the Copenhagen Emergency Services to develop a system for recognizing stroke cases in calls to the 1-1-2 emergency line and the 1813 medical helpline. Stroke is one of the biggest causes of disability and death worldwide [199, 329, 360] and effective treatment is highly time-sensitive [46, 652]. The most common gateway to specialized treatment and hospital admittance is through prehospital telehealth services like emergency medical call centers, nurse advice call lines, and out-of-hours health services [236, 476], however, studies have found that approximately half of all patients with stroke do not receive the correct triage for their condition from call-takers [54, 502, 668]. This is likely due to the complexity of stroke cases which exhibit a wide range of symptoms that can be difficult to recognize over the phone. Additionally, stroke cases are relatively rare, occurring in only 0.25% of all calls made to the Copenhagen 1813 medical helpline in 2021. This makes it difficult for call-takers to gain practical experience with stroke cases which compounds to the fundamental difficulty

of recognizing them. Although several efforts have been made to improve recognition rates, there is still a need for better tools to support call-takers in recognizing stroke cases [52, 53, 696].

Uncertainty in stroke recognition   A reasonable machine learning pipeline to assist in recognizing stroke cases on calls to emergency services could consist of an automatic speech recognition model that transcribes the conversation and a binary classifier that estimates the probability that the transcript describes a case of stroke. Such a system might be trained on a dataset calls with verified stroke and non-stroke cases and evaluated against a held-out test set of calls where the call-taker has indicated whether they suspect a stroke. Due to the low prevalence of stroke cases, the dataset would be highly unbalanced, and the number of stroke cases limited. In such a system, many factors can lead to classification errors, and when that happens, we would like the model to express high degree of uncertainty. For instance, the automatic speech recognition model might make an error in the transcription of particular word or phrase due to a noisy environment, overlapping, slurred or mumbling speech, or words not in the model's training data. The classifier might misclassify a conversation due to medically ambiguous symptoms, a general lack of information given in the conversation, or transcription errors made by the speech recognizer. How to best make such models accurately represent and estimate the uncertainty of given predictions, and how to use it to improve the value of such a system, especially in cooperation with the call-taker, is an open question. What is clear, however, is that the ability of such a system to accurately estimate the uncertainty of given predictions is crucial for its safe and reliable deployment in the real world. Overconfident predictions could lead to unnecessary delays in treatment, misdiagnoses, increased costs for the healthcare system, and potentially fatal consequences for patients.

1.1.2   Medical coding of clinical notes

Case background   During the course of this project, Corti's portfolio has expanded to include a software system for medical coding of clinical notes. When a patient is admitted to a hospital, the medical staff will write a clinical note describing the patient's diagnosis and the procedures they underwent, including drug prescriptions. These notes are then used for billing and reimbursement purposes, as well as for research and quality assurance. The clinical notes are written, usually by a doctor, in natural language adhering to a certain structure. Later, a medical coder will assign a set of codes to the note based on its content. The process of medical coding is time-consuming and error-prone due to the vast number and high complexity of medical diagnoses and procedures. For instance, the widely used International Classification of Diseases (ICD) standard consists

of 55,000 medical codes in version 10 and 85,000 in version 11 [707]. Additionally, a single clinical note will usually contain several diagnoses and procedures which must all be inferred from the natural language text [314, 315]. Furthermore, any single code can have several criteria defined by official guidelines that determine under which conditions it is mutually exclusive with other codes, and when other codes must be coded along with it [290]. These properties make medical coding a very complex, multi-label classification problem.

Uncertainty in automated medical coding   A reasonable machine learning pipeline to assist in medical coding could consist of a natural language processing model that extracts relevant information from the clinical note and outputs a probability distribution for each of the potential medical codes. However, training and using such as system comes with several challenges. For instance, the prevalence of different medical codes in the training data is highly unbalanced; it is common to have several orders of magnitude difference between the frequency of the most frequent code and the least frequent code. Since each clinical note is associated with multiple codes that have complex co-occurrence patterns, it is often impossible to exactly correct for this class imbalance by stratified sampling, which is the usual go-to approach to deal with class imbalance in machine learning. Furthermore, some codes are highly similar, for instance the ICD-10 codes Z87.891 "Personal history of nicotine dependence" and F17.210 "Nicotine dependence, cigarettes, uncomplicated". In practice, however, only one of them should be assigned to a given note. As with the stroke recognition system, we would like the model to express high degree of uncertainty when it makes a mistake. For example, if the model is uncertain about two similar codes, it might be appropriate to ask a human expert to review the note and make the final decision. Additionally, the lack of training data for the rarest codes makes them difficult to learn to robustly predict, and so, we might reasonably expect the model to often be uncertain for rare codes.

1.2   MACHINE LEARNING RELIABILITY

Several factors define the reliability of a machine learning model. Besides model performance and accuracy, important factors include the *interpretability* of how the model functions, the *explainability* of its predictions, *fairness* in its treatment of different groups, and *robustness* to noise, outliers, and adversarial attacks. Since many modern machine learning models are deep neural networks with millions or billions of parameters, their size and complexity make them inherently difficult to interpret, and their predictions hard to explain. Due to high cost and practical infeasibility, the vast amounts of data needed to train such models are often not manually curated for quality, and so, may contain biases and errors

which models are well-equipped to learn to mirror, risking fairness [71]. Finally, a number of factors, including the ability of deep neural networks to overfit, or even memorize, their training data [12, 70], can lead to models that are sensitive to adverse noise conditions and outliers.

The ability of a model to accurately estimate the uncertainty of its predictions plays an important role in its reliability. While associating any prediction with an accurate uncertainty estimate can be argued to improve both interpretability, explainability, and fairness, the main goal is to improve the model's robustness to noise, outliers, and adversarial attacks. Given an input which cannot be mapped to a single output with certainty, the model can then indicate that its prediction should not to be trusted. However, overfitting, memorization, and the use of training objectives that are proxies for the evaluation metrics of interest often lead to models that are miscalibrated; that is, the predicted probability of a class does not reflect the true probability of the model being correct [226, 358]. Usually, the predicted probabilities are too extreme which leads to overconfident predictions. This means a model will often assign a high probability to the predicted class, even when it is wrong.

1.2.1   MODEL CALIBRATION

The problem of miscalibration in machine learning models is well-known and has been studied for decades [44, 195, 394, 489, 528, 733]. One of the best known methods for calibrating a binary classifier is Platt scaling which fits a logistic regression model to the model outputs, assuming that the miscalibration can be corrected by a logarithmic function [528]. Another method is isotonic regression which instead fits a nonparametric, monotonic function [733]. More recently, Guo et al. [226] proposed a single-parameter variant of Platt-scaling that fits only a temperature parameter on the logits of a neural network classifier. These methods are generally simple and effective, but not without limitations. To perform the calibration, most of the methods require a held-out validation set on which the model was not trained, Platt scaling and isotonic regression are not directly applicable to multi-class classification problems, and how to calibrate models for structured prediction tasks, such as speech recognition and machine translation, remains an open problem [15, 16, 302].

More importantly, correct calibration of a machine learning model does not guarantee that the predicted probability is accurate. Specifically, even a well-calibrated model can be overconfident for data that were not presented to it during training such as rare events, outliers, and adverse examples, sometimes collectively referred to as out-of-distribution examples. For instance, take a perfectly calibrated model trained to classify images of cats and dogs. If presented with an image of a horse, the model has no option but to distribute 100% total proba-

bility across the cat and dog categories, even though that is clearly wrong. Worse yet, to indicate maximal uncertainty, we might expect the model to assign 50% probability to each of the cat and dog categories, but sadly that behavior is not guaranteed. On the contrary, since no horses were in the training data, the model will not have learned specialized features for horses, nor learned to associate any relevant known features with horses. So, if a particular horse has features that resemble those learned for a dog more than for a cat, the model might assign arbitrarily high probability to the dog class; and vice versa. Therefore, even perfectly calibrated models risk being confidently wrong [745].

1.2.2 UNDERSTANDING UNCERTAINTY

As hinted at in the previous section, there are different sources of uncertainty in machine learning models. At a fundamental level, we can decompose the *predictive uncertainty* into uncertainty that is present in the knowledge we have, and uncertainty that originates from the knowledge that we do not have. These sources are sometimes called *known unknowns* and *unknown unknowns*, or referred to in terms of *aleatoric uncertainty* and *epistemic uncertainty* [333]. Aleatoric comes from the Latin word 'aleatorius' for 'dice player' or 'gambler' and refers to uncertainty present in the data itself due to randomness in the process that generated it. This kind of uncertainty is commonly seen as irreducible but, since it is represented in the collected data, it can usually be modelled directly. Epistemic comes from the Greek word 'epistēmē' which means 'knowledge' and refers to uncertainty due to things one could in principle know, but does not in practice. Such lack of knowledge could for instance be due to a lack of data, or an improper model specification. Aleatoric uncertainty is sometimes referred to as *stochastic uncertainty* and epistemic uncertainty as *systematic uncertainty* [333].

An example of aleatoric uncertainty is the uncertainty in the transcription of a word due to a noisy environment or overlapping, slurred or mumbling speech. Unknown words are arguably sources of epistemic uncertainty although a good speech recognition model might generalize well if the word's spelling and pronunciation follow the same patterns as in the training data. Another example of epistemic uncertainty is the occurrence of truly out-of-distribution examples such as the horse in the image classifier example from earlier: inputs unlike anything the model has seen during training, especially if they require outputs unavailable to the model. In high-dimensional data with a practically unlimited diversity in out-of-distribution examples, it is impossible to collect enough data to completely eliminate all potential sources of epistemic uncertainty. By including horses in the training data, the model would likely learn to recognize them, but would still be unable to recognize giraffes, or zebras, or unicorns. Ultimately, we must accept that any practical machine learning model will have some sources

of epistemic uncertainty.

As mentioned, aleatoric uncertainty is represented by the data, and so can usually be modelled directly. For instance, in the case of speech recognition, we can model the uncertainty in the transcription of a word by a distribution over the words in the vocabulary. But how do we model epistemic uncertainty? Since sources of epistemic uncertainty are by definition those not represented in the data, it is generally not possible to model them directly. This makes epistemic uncertainty more difficult to quantify than aleatoric uncertainty. Methods for estimating epistemic uncertainty include Bayesian probability and Bayesian neural networks [438, 479] which represent it by an explicit distribution over learned model parameters. Ensemble methods [191, 368] take a similar approach but use an implicit distribution. Other approaches include anomaly detection and the recent field of out-of-distribution detection which, for example, represent epistemic uncertainty by special output classes for out-of-distribution data, distributions over data representations, or distances between them (see section 3.2). In this thesis, we will focus on quantifying epistemic uncertainty using out-of-distribution detection.

1.3   Thesis scope and outline

In this introduction we provided a high-level overview of the motivation behind the research project by focusing on the use-cases of stroke recognition and automated medical coding, as well as speech processing and the importance of uncertainty quantification. The remainder of part I consists of two chapters. Chapter 2 presents the research hypotheses and contributions of the thesis via the included papers. Chapter 3 provides relevant technical background to the included papers, including uncertainty, out-of-distribution detection, and variational autoencoders.

Parts II to IV contain a number of chapters that each correspond to a primary paper. Part II is made up of two papers that deal with out-of-distribution detection using generative models, including variational autoencoders [45, 245]. Part III consists of two papers that explore speech representation learning with variational autoencoders and self-supervised methods [58, 244]. Part IV consists of two papers on applications of machine learning within the medical domain: medical coding of clinical notes [171] and recognition of stroke cases in calls to medical helplines [698]. Finally, part V concludes the thesis by discussing the presented work and future directions for research.

RESEARCH HYPOTHESES AND CONTRIBUTIONS

The chapters of parts II to IV are self-contained studies and therefore detail their own research hypotheses and contributions. Since each study was written without consideration to the other chapters of this thesis, we here detail them in relation to the overall research project. This constitutes the research hypotheses and the contributions of the thesis.

PART I
BACKGROUND
Chapter 1 provides a general introduction to the thesis and gives motivating examples for speech recognition and assistance in medical encounters including stroke recognition on medical helplines and automation of medical coding. Chapter 3 provides additional technical background not included in the individual studies. It introduces uncertainty as a concept in the context of information and probability theory and introduces the task of out-of-distribution detection and provides a review of existing work on the problem. Finally, it lays out the foundations for variational autoencoders.

PART II
UNSUPERVISED UNCERTAINTY ESTIMATION
This part of the thesis is concerned with unsupervised uncertainty estimation and consists of two papers. Both papers focus on using generative models for out-of-distribution detection, which is the task of detecting data that are likely to be sampled from a different data generating distribution than the training data. In both cases, the contributions are methodological and relate to developing improved methods for out-of-distribution detection.

CHAPTER 4
HIERARCHICAL VAES KNOW WHAT THEY DON'T KNOW
In this work we hypothesize that the likelihood estimate of variational autoencoders is a poor score for out-of-distribution due to an overemphasis on low-level features that generalize between distributions. We further hypothesize that a well-formed hierarchy of latent variables provides a tool that can be used to select which features to emphasize for out-of-distribution detection and, hence, a way to improve the performance of variational autoencoders on this task. We proceed to provide empirical and theoretical evidence that low-level features do indeed dominate the likelihood score and propose a new method for out-of-distribution detection using hierarchical variational autoencoders based on a

likelihood-ratio score that requires data to be in-distribution across all feature-levels. The proposed method is computationally efficient, fully unsupervised, and performs well on several out-of-distribution detection benchmarks.

CHAPTER 5

MODEL-AGNOSTIC OUT-OF-DISTRIBUTION DETECTION USING COMBINED STATISTICAL TESTS

In this follow-up work to chapter 4, we note that the set of methods proposed for out-of-distribution detection using generative models is quite large and that many are tailored for specific model types, which suggests that it is possible to develop a model-agnostic approach. We hypothesize that by phrasing the task as a statistical testing problem and combining different tests, the method's efficacy can be improved and weaknesses inherent to any particular test can be alleviated. From this hypothesis, we combine a classical parametric test with the recently introduced typicality test to develop a method applicable to any differentiable generative model with explicit likelihood, and show that this leads to a more accurate out-of-distribution test. Finally, we discuss the benefits of casting out-of-distribution detection as a statistical testing problem, for instance enabling false positive rate control. This property is valuable in many practical applications, especially in high-risk settings such as medical decision-making.

PART III

UNSUPERVISED SPEECH REPRESENTATION LEARNING

This part deals with unsupervised learning of speech representations and consists of two papers. Speech representations are fundamental to any practical system for decision support as well as for uncertainty quantification on speech data. The contributions of this part lie in analyzing and comparing different approaches to speech representation and in providing a comprehensive overview of the field.

CHAPTER 6

A BRIEF OVERVIEW OF UNSUPERVISED NEURAL SPEECH REPRESENTATION LEARNING

In this chapter, we present a comprehensive overview of unsupervised neural representation learning for speech. Previous research is categorized into self-supervised methods and probabilistic latent variable models and described in a common notation. This description assists in developing a model taxonomy that shapes a discussion of the models' representational power, the associated learning strategies, and the methods used to evaluate them. The discussion points to interesting avenues of future research. An extended version of this overview paper that focuses exclusively on self-supervised methods was also published as part of the project [G] [461] and is included in appendix A for reference.

CHAPTER 7
BENCHMARKING GENERATIVE LATENT VARIABLE MODELS FOR SPEECH
This chapter develops a novel hierarchical latent variable model for speech, drawing inspiration from the Clockwork VAE [577]. A comparative benchmark against alternative latent variable models and autoregressive models for speech highlights the improvements to likelihood brought by using hierarchical latent variables. The paper also analyzes the latent spaces learned by the models in terms of phonetic content.

PART IV
MEDICAL APPLICATIONS
This part contains two studies on machine learning methods applied for tasks in a medical setting. The contributions of the first are methodological focusing on improved comparability and reproducibility of studies on automated medical coding. The second is a retrospective study on machine learning-assisted stroke recognition focusing on the potential clinical impact of using machine learning to recognize stroke cases in emergency calls. While uncertainty estimation is not a central theme to the two papers, the retrospective study performs a substantial evaluation of the explainability of the proposed model via an occlusion analysis on the text input. Later, in the discussion, we further consider uncertainty estimation in relation to medical applications focusing on the retrospective study (section 10.3).

CHAPTER 8
AUTOMATED MEDICAL CODING ON MIMIC-III AND MIMIC-IV: A CRITICAL REVIEW AND REPLICABILITY STUDY
In this paper, we review the current state-of-the-art in automated medical coding of clinical notes. We hypothesize that several previous works underperform for reasons more related to suboptimal hyperparameter tuning, incorrect evaluation, and crude data handling than to model design, and that performance and comparability can be improved by addressing these issues. We first reproduce, analyze and compare several models on the MIMIC-III dataset showing that poor performance is indeed attributable to weak configuration of the training and poorly sampled train-test splits with many extremely rare classes - several without examples in the training data. We also identify and correct a widespread error in the calculation of the macro F1-score. To compare models, we propose new data splits created with stratified sampling, use identical experimental setups and tune hyperparameters and decision boundaries. By analyzing prediction errors, we confirm the observation of previous work that all models struggle with rare codes, although, contrary to previous claims, long documents only have a negligible impact on performance. Finally, we present the first comprehensive

results on the recently released MIMIC-IV dataset using the reproduced models.

CHAPTER 9

A RETROSPECTIVE STUDY ON MACHINE LEARNING-ASSISTED STROKE RE-COGNITION FOR MEDICAL HELPLINE CALLS

In this paper we examine the hypothesis that a machine learning framework can learn to recognize cases of stroke in calls made to a prehospital medical helpline. We used calls from Copenhagen during 2015 to 2020, to develop a machine learning-based classification pipeline. First, calls were transcribed by a speech recognition model and then categorized as stroke or non-stroke using a text classification model. On test data from 2021, call-takers achieved an overall sensitivity of 52.7% (95% confidence interval 49.2-56.4%) with a positive predictive value (PPV) of 17.1% (15.5-18.6%) while the machine learning framework performed significantly better ($p < 0.0001$) with a sensitivity of 63.0% (62.0-64.1%) and a PPV of 24.9% (24.3-25.5%). Effective treatment of out-of-hospital stroke often hinges on recognition by call-takers at prehospital telehealth services. This study provides preliminary evidence that a machine learning framework could become a supportive tool for call-takers at prehospital medical helplines, aiding in early and accurate stroke recognition.

TECHNICAL BACKGROUND

While the papers in this thesis were written to be self-contained, page limits and the need to maintain a specific focus forced us to limit the scope of the technical background that each paper provides. In this chapter, we present a more in-depth introduction to the technical background relevant to the papers in this thesis. We start by quantifying the concept of uncertainty and relating it to information and probability. We then introduce the task of out-of-distribution detection and review the existing literature on the topic. Finally, we describe variational autoencoders, how they can be used to learn representations of speech, and some of their challenges.

3.1 UNCERTAINTY AND INFORMATION

In society, the concept of uncertainty goes by many names, and its meaning can vary depending on the specific context. However, across most quantitative scientific fields, a consensus definition appears to align with the following [287]:

> *Uncertainty is the lack of certainty; a state of limited knowledge where it is impossible to exactly describe the existing state, a future outcome, or more than one possible outcome.*

Although such a purely lexical definition of uncertainty might prompt philosophical inquiry, it highlights an important connection between uncertainty, information and probability. When information is limited, we can describe the state of the world only with uncertainty; we are simply not certain. In that sense, uncertainty is a word we use to refer to missing information, whether we know what is missing or not. A natural way to describe an uncertain world state is to use probabilities. To our luck, information theory provides exactly that; a mathematically rigorous method for quantifying information through the language of probability theory. In the following, we shall see that, in this context, uncertainty can be understood as the information entropy of a random variable [439].

3.1.1 ENTROPY FOR DISCRETE RANDOM VARIABLES

First characterized by Claude Shannon in 1948 [596], information entropy, $H(X)$, is a measure of the average amount of information contained in a discrete random variable X. Shannon's definition follows from three fundamental axioms of information theory. Let $I_X(x)$ be the information carried by a specific outcome x

of sampling the discrete random variable X with probability mass function $p_X(x)$. Then, these axioms are as follows:

(I) The more likely an outcome is, the less information it carries; $I_X(x)$ is a monotonically decreasing function in $p_X(x)$.

(II) Outcomes that are certain to happen carry no information; if $p_X(x) = 1$ then $I_X(x) = 0$.

(III) The joint information carried by independent outcomes is the sum of the information carried by each outcome; if x_i and x_j are independent then $I(x_i, x_j) = I(x_i) + I(x_j)$.

From these axioms, Shannon found that the *information content* $I_X(x)$ of an outcome x with probability $p_X(x)$ can suitably be defined as the negative logarithm of the probability of the outcome,

$$I_X(x) = -\log_b p_X(x) \ . \tag{3.1}$$

The information entropy of a discrete random variable X is defined as the *expected information content* of an outcome of X,

$$H(X) = \mathbb{E}_X \left[I_X(x) \right] = -\sum_{x \in X} p_X(x) \log p_X(x) \ . \tag{3.2}$$

This means that $H(X)$ measures the amount of information we can expect to gain from observing an outcome of X, when we know only its distribution $p_X(x)$. If all possible outcomes are equally likely, then $H(X)$ is maximized, and if only one outcome is possible, then $H(X)$ is minimized.

3.1.2 ENTROPY FOR CONTINUOUS RANDOM VARIABLES

To generalize the concept of information entropy to continuous random variables, Shannon originally replaced the sum over the probability mass function in (3.2) with an integral over the probability density function, as suggested by the definition $H(X) = \mathbb{E}_X \left[I_X(x) \right]$. This approach leads to the definition of the information entropy of a continuous random variable X as,

$$H(X) = -\int_{\mathcal{X}} p_X(x) \log p_X(x) \, \mathrm{d}x \ , \tag{3.3}$$

which is called the *differential entropy* [596]. Here, $\mathcal{X}$ is the support of X.

However, the differential entropy does not have several of the desirable properties of the discrete version; it can be negative, it is not invariant under a change

of variables, and it is not dimensionally correct.[1] For these reasons, the differential entropy may not be a suitable measure of information, or uncertainty, for continuous random variables.

Instead, Jaynes [303, 304] argued that the information entropy of a continuous distribution should be defined as the limiting density of increasingly dense discrete distributions, $H_{N \to \infty}(X)$. This argument leads to,

$$H_{N \to \infty}(X) \equiv \lim_{N \to \infty} [H_N(X) - \log N] = - \int_{\mathcal{X}} p_X(x) \log \left(\frac{p_X(x)}{q_X(x)} \right) \, dx \ , \qquad (3.4)$$

where $H_N(X)$ is the limiting density for discrete points, $q_X(x)$ is a uniform density over the quantization of the continuous space, and we have subtracted a $\log N$ term that would go to infinity in the limit of infinite points. By doing this, the information entropy becomes positive, dimensionally correct, and invariant under a change of variables.[2]

The form on the right-hand side of (3.4) can be recognized as the negative Kullback-Leibler divergence of $p_X(x)$ to $q_X(x)$ [359],

$$D_{KL}(p_X(x) \parallel q_X(x)) = \int_{\mathcal{X}} p_X(x) \log \left(\frac{p_X(x)}{q_X(x)} \right) \ . \qquad (3.5)$$

The Kullback-Leibler divergence $D_{KL}(p_X(x) \parallel q_X(x))$, also known as the *information gain*, is often interpreted as the amount of additional information required to

---

[1] Information entropy has the same dimensionality as information content which is typically measured in units of bits, but depends on the base of the logarithm. However, the bit is not a base unit of the International System of Units (SI), nor is it an official derived unit. A bit is quite simply a number, specifically 0 or 1. If something is said to have a size of 4 bits, it means that it can be described with 4 binary digits, i.e. 4 numbers, each either 0 or 1.

As such, it might be useful to think of information as a dimensionless quantity rather than a quantity with units. Following this interpretation, the discrete information entropy in (3.2) is also dimensionless, because a probability mass function is dimensionless. Since probability density functions have dimensionality of the inverse of some quantity, e.g. inverse length, the dimensionality of the continuous differential entropy in (3.3) becomes,

$$\dim (H(X)) = \dim \left(\int_{\mathcal{X}} p_X(x) \log p_X(x) \, dx \right) = \frac{1}{\text{length}} \log (\text{length}) \ ,$$

which is clearly not dimensionless and therefore cannot correspond directly to the discrete information entropy.

[2] Under a change of independent variable from x to $y(x)$, we have that

$$\widetilde{w}(y) \, dy = w(x) \, dx \ , \quad \widetilde{q}(y) \, dy = q_X(x) \, dx \ .$$

Plugging this into (3.4) we arrive at the original expression but now with y as the independent variable,

$$H_{N \to \infty}(y(x)) == - \int_{\mathcal{Y}} \widetilde{w}(y) \frac{dy}{dx} \log \left(\frac{\widetilde{w}(y) \frac{dy}{dx}}{\widetilde{q}(y) \frac{dy}{dx}} \right) \, dx = - \int_{\mathcal{Y}} w(y) \log \left(\frac{w(y)}{q(y)} \right) \, dy \ .$$

represent events from a distribution with density $p_X(x)$ using a code optimized for a distribution with density $q_X(x)$. In other words, it measures how surprised one would be if they used distribution q to represent events from distribution p, i.e. the *relative entropy* of p with respect to q.

Returning to Jaynes' argument and (3.4), we can interpret this to say that the information entropy of a continuous random variable X should be defined as the expected difference in information entropy between its density and a uniform density. Identifying the Kullback-Leibler divergence as a measure of this relative information entropy provides a natural point of convergence for this section since it is equivalently defined for both discrete and continuous random variables, contrary to the differential entropy. In this view, information entropy is a quantity that distills the distribution of a random variable into a single number describing the diversity of the potential outcomes of the random variable.

The Kullback-Leibler divergence plays a central role in the training of variational autoencoders, which we introduce in section 3.3. In chapter 4 we shall see how the Kullback-Leibler divergence emerges in a likelihood-ratio test statistic for out-of-distribution detection using hierarchical variational autoencoders. In the remainder of the thesis, we will write the density of a random variable X as $p(x)$, dropping the subscript, as is usual in machine learning.

3.2   OUT-OF-DISTRIBUTION DETECTION

In section 3.2.1 we introduce out-of-distribution (OOD) detection and connect it to the concept of uncertainty. Since the related work sections of the papers in chapters 4 and 5 are relatively short, we also provide a concise review of the literature on out-of-distribution detection in section 3.2.2 and 3.2.3.

3.2.1   AN OVERVIEW OF THE TASK

OOD detection is the task of identifying test data that are unlikely to originate from the distribution of the training data and, in the context of neural networks, dates back several decades [50, 87]. In the general case, we assume that we have a domain of in-distribution data, $\mathcal{D}_{in}$, and we would like to build a model that can be used to assess whether a test data point **x** originates from that domain or not. Since OOD data is unknown to the model, it is, by definition, a source of epistemic uncertainty. This makes the task of OOD detection an instance of uncertainty quantification.

Related tasks   OOD detection bears many similarities with anomaly detection, novelty detection, open set recognition, and outlier detection [721]. Some of these differences are subtle, and the terminology is not always used consistently

in the literature. To provide some clarity, we will briefly outline the taxonomy of Yang et al. [721]: Outlier detection most clearly differs from the other tasks by aiming to select outliers from a single contaminated dataset by processing all observations together. The remaining tasks differ in whether they detect both covariate and semantic shift and require the simultaneous classification of in-distribution classes. Anomaly detection deals with multiclass data and detects both covariate and semantic shift without requiring simultaneous classification of in-distribution classes whereas novelty detection, open set recognition and OOD detection are usually concerned only with semantic shifts. Although a vague difference, novelty detection is usually defined in terms of a class of normal data, while OOD detection centers around the distribution of the training data. Different from novelty detection, OOD detection methods sometimes draw on examples of OOD data or require the simultaneous classification of in-distribution classes, although this is not always the case. OOD detection benchmarks almost always take OOD data to be from external datasets different from the training dataset which distinguishes it from open set recognition that usually uses a single dataset split into ID and OOD data. Although we share the sentiment of Yang et al. [721], who propose to unify all these tasks as "generalized out-of-distribution detection", this thesis will follow the nomenclature used in the works most related to ours and refer to the task as out-of-distribution detection.

Supervision Recently, several approaches for deep neural networks have been developed that address the rejection of OOD samples $\mathbf{x} \notin \mathcal{D}_{\mathrm{in}}$. One way to categorize the different approaches is based on whether the underlying model is a classifier that estimates a conditional probability distribution $p(y|\mathbf{x})$ over some target variable y, or a model that learns a probability density $p(\mathbf{x})$ over the input itself. We might refer to the former as *supervised* OOD detection, and the latter as *unsupervised* OOD detection [215, 423]. It is important to note that this distinction relates only to the availability of some target value, y – not whether OOD data is available for supervision.[3] Supervised OOD detection tries to assess whether the model's prediction $\hat{y}$ via $p(y|\mathbf{x})$ should be trusted for a given input $\mathbf{x}$, whereas unsupervised OOD detection judges whether the input $\mathbf{x}$ should be trusted and used at all.

---

[3]In our overview, we distinguish between supervised and unsupervised out-of-distribution by letting methods be classified as supervised that use any kind of target value y, whether it relates to an original (in-distribution) task, available OOD data, or both. This is the same distinction made by Graham et al. [215] and Liu et al. [423]. It is important to note, however, that in other works, the distinction is made based on whether the model is trained on OOD data or not [253, 417]. This difference in nomenclature is currently unresolved in the literature.

Scoring Central to any OOD detection method is the ability to assign a score $s(\mathbf{x}) \in \mathbb{R}$ to a given input $\mathbf{x}$ to indicate the degree to which the input is likely to be OOD. After defining a score, we typically use a validation set to tune a threshold τ such that $\mathbf{x}$ is considered OOD if $s(\mathbf{x}) > \tau$. The threshold is typically chosen such that performance on the validation set is above some level, for instance by imposing constraints on the recall and precision. Many works also evaluate the performance of OOD detection methods using the area under the receiver operating characteristic (AUROC) curve which does not require a threshold to be chosen.

There are several general ways to define a score $s(\mathbf{x})$, and they can be used to further categorize OOD detection methods. Unsupervised methods often derive the score from the *likelihood* assigned to the input [45, 50, 107, 245, 347, 434, 467, 473, 552, 592, 716], from a *reconstruction* of the input [95, 215, 397, 423, 431, 571, 582, 712, 742, 749], or from a hidden *representation* of the input [2, 47, 150, 255, 590, 627, 715]. Supervised methods often derive the score from the *probabilities* given by the predictive distribution $p(y|\mathbf{x})$ [252, 253], from the *logits* of $p(y|\mathbf{x})$ [252, 417], or use a hidden *representation* of the input [133, 389, 396, 478, 734], similar to unsupervised methods. In the following sections, we provide a more in-depth overview of the different approaches to OOD detection following the above categorization into supervised and unsupervised methods. For the supervised methods we further distinguish between whether methods use real OOD data, synthetic OOD data, or no OOD data at all.

3.2.2 SUPERVISED OUT-OF-DISTRIBUTION DETECTION

In the supervised setting, we take the supervision from the target variable y to be unrelated to the nature of the OOD data. However, a number of works on supervised OOD detection use real or synthetic OOD data to augment the training objective of the classifier $p(y|\mathbf{x})$ to better detect OOD data. For that reason, in the following, we will categorize the supervised OOD methods depending on whether they require the use of OOD data or not, and if they do, whether the used OOD data is real or synthetic.

Methods using real OOD data Methods that use representative OOD data have achieved high performance since they can directly learn to distinguish between in-distribution and OOD data. Hendrycks, Mazeika, and Dietterich [254] augment the original training objective with a task-dependent outlier exposure loss that aims to make the output logits discriminative of outlier data. In a similar vein, Dhamija, Günther, and Boult [153] propose losses designed to maximize the entropy of $p(y|\mathbf{x})$ and decrease feature magnitudes for OOD data sampled from other datasets. Ruff et al. [569] use semi-supervised learning and learn rep-

resentations of in-distribution data that concentrate close to a centroid in latent space, while labeled outliers are pushed away from the centroid. Other methods including MCD [728], NGC [710], and UDG [720] use external, unlabeled, noisy data to improve OOD detection performance without requiring cleanly labeled OOD examples.

Methods using synthetic OOD data   A number of methods do not require access to actual OOD data but synthesize it instead. Several methods do so by adding noise to in-distribution data [389, 400, 552]. For instance, Ren et al. [552] propose a number of baselines including training a binary classifier to distinguish between original and perturbed in-distribution data. They also propose adding an OOD class to softmax classifiers and training it to predict perturbed in-distribution data, or alternatively, training the predicted class distribution to output uniform distribution for perturbed in-distribution inputs. While these methods are appealing, their weaknesses have been pointed out by later work which generally improve on their performance. An example is ODIN [400], in which the authors propose to calibrate $p(y|\mathbf{x})$ with temperature scaling [226] and add gradient-based, input-dependent perturbations to the inputs and use the calibrated maximum class probability as the OOD score. Vyas et al. [672] train an ensemble of classifiers on different subsets of the training data, with the left out data taken as OOD, and propose novel loss over $p(y|\mathbf{x})$ that seeks to maintain a predefined margin between its average entropy for the OOD and in-distribution examples. Another approach generates OOD inputs using a generative adversarial network [388].

Methods not using OOD data   A group of methods do not rely on OOD data for accurate detection. An early baseline approach uses the maximum class probability of $p(y|\mathbf{x})$ directly by noting that it tends to be larger for correctly classified examples [253]. Another baseline method proposes that a high entropy of $p(y|\mathbf{x})$ indicates an OOD input [552]. Other methods that derive the score from the classifier probabilities include that of Lakshminarayanan, Pritzel, and Blundell [368] who propose to use an ensemble of independently trained classifiers to discriminate between in-distribution and OOD data by evaluating the agreement between the classifiers, but also DeVries and Taylor [152] who augment the network with a confidence estimation branch that learns to estimate the confidence of the classifier separately from the probability, and Huang, Geng, and Li [283] who compute the gradient of the KL-divergence of the predictive distribution $p(y|\mathbf{x})$ to a uniform distribution noting that the magnitude of gradients is higher for in-distribution data than for OOD data. Huang and Li [284] group the classes of the target variable y and define an OOD class for each group. Each training example is then the correct target for one group and an OOD example for all other

groups; a kind of hierarchical version of the OOD class of simpler baselines based on noise augmentation [552]. The Variational Information Bottleneck [6] jointly learns a probabilistic latent representation, $p(\mathbf{z}|\mathbf{x})$, and a classifier, $p(y|\mathbf{x})$, using a generalized variational autoencoder and tries to maximize the mutual information between $\mathbf{z}$ and y and minimize it between $\mathbf{z}$ and $\mathbf{x}$.

Hsu et al. [280] propose a generalized version of ODIN that removes the need for simulating OOD data. The authors note that most current methods make a closed world assumption and implicitly condition on the in-domain $\mathcal{D}_{in}$ in the form of the predictive distribution $p(y|\mathbf{x}, \mathcal{D}_{in})$. With this observation, the authors decompose the $p(y|\mathbf{x}, \mathcal{D}_{in})$ into a joint class-domain probability and a domain probability,

$$p(y|\mathbf{x}, \mathcal{D}_{in}) = \frac{p(y, \mathcal{D}_{in}|\mathbf{x})}{p(\mathcal{D}_{in}|\mathbf{x})} \quad . \tag{3.6}$$

Without data from the out-domain, it is not possible to directly learn either $p(y, \mathcal{D}_{in}|\mathbf{x})$ or $p(\mathcal{D}_{in}|\mathbf{x})$. Instead, the authors use this observation to impose the inductive bias of predicting logits as a fraction between two carefully designed network branches, imitating the form of (3.6).

Although some works use the maximum softmax probability as a score for OOD detection [254, 552], several works have noted that it is not generally reliable [252, 417]. Liu et al. [417] make an interesting argument as to why based on the energy $E(\mathbf{x}; f)$ of a softmax classifier $f(\mathbf{x})$ [378],

$$E(\mathbf{x}; f) = -\log \sum_{i=1}^{K} \exp(f_i(\mathbf{x})) \quad . \tag{3.7}$$

Specifically, the authors write the maximum softmax probability as,

$$\begin{aligned}
\max_y p(y|\mathbf{x}; f) &= \max_y \frac{\exp f_y(\mathbf{x})}{\sum_{i=1}^{K} \exp f_i(\mathbf{x})} \\
&= \frac{\exp f^{max}(\mathbf{x})}{\sum_{i=1}^{K} \exp f_i(\mathbf{x})} \\
&= \frac{1}{\sum_{i=1}^{K} \exp(f_i(\mathbf{x}) - f^{max}(\mathbf{x}))} \quad ,
\end{aligned} \tag{3.8}$$

where $f^{max}(\mathbf{x}) = \max_i f_i(\mathbf{x})$. The authors then relate the maximum softmax probability to the energy by noting that,

$$\log \max_y p(y|\mathbf{x}; f) = E(\mathbf{x}; f(\mathbf{x}) - f^{max}(\mathbf{x})) = E(\mathbf{x}; f) - f^{max}(\mathbf{x}) \quad . \tag{3.9}$$

This shows that log of the softmax confidence score is equivalent to the special case of the energy score where all the logits are shifted by their maximum logit

value. The authors empirically observe that the energy $E(\mathbf{x}; f)$ tends to be larger for OOD data than for in-distribution data, while $f^{max}(\mathbf{x})$ tends to be smaller. They conclude that this shift results in the maximum softmax probability being a biased score for OOD detection and propose to instead use the energy directly. This energy-score was further improved in ReAct by feature clipping [622].

The final category of supervised methods derive the score from a hidden representation of the input. For instance, to represent a virtual OOD class, Wang et al. [681] generate an additional logit by first computing the residual of the input's latent space representation against a principal feature space and then converting it to a valid logit by matching its mean over training samples to the average maximum logits. Other methods note that the difficulty of detecting OOD data might be attributed to the curse of dimensionality in the learned feature spaces and propose to use dimensionality reduction techniques. Ndiour, Ahuja, and Tickoo [478] apply dimensionality reduction on learned, high-dimensional features to capture the true feature subspace and compute the norm of the difference between the original feature and the pre-image of its low-dimensional manifold embedding. Zaeemzadeh et al. [734] force the ID samples to embed into a union of 1-dimensional subspaces during training and computes the minimum angular distance from the feature to the class-wise subspaces. NuSA [133] projects features onto the column space of the classification weight matrix and computes the ratio of the norm the projected and original features. Lee et al. [389] fit a multivariate Gaussian distribution to the activations of the penultimate layer of a pre-trained classifier and use the Mahalanobis distance to this distribution to evaluate whether inputs are OOD. This method can also be seen as ameliorating the curse of dimensionality by clustering the high-dimensional feature space. Finally, Bayesian neural networks have also been proposed for OOD detection, but their performance is not yet competitive with other methods [141, 256, 484].

A note on limitations As we discussed earlier, for modern applications of machine learning, input data is often high dimensional and complex which makes it difficult to obtain enough representative OOD data to ensure robust OOD detection capabilities. This inescapable property limits the practical usefulness of methods that use real OOD data, as well as those that synthesize it. Furthermore, a general weakness of all supervised out-of-distribution detection is that in learning the task-specific model $p(y|\mathbf{x})$ a model may discard information about $p(\mathbf{x})$ which could be useful for out-of-distribution detection. While augmenting training objectives and (re)training models specifically for OOD detection might alleviate this issue, it does not remove the issue.

3.2.3   UNSUPERVISED OUT-OF-DISTRIBUTION DETECTION

For high-dimensional data, the most successful unsupervised models for OOD detection are deep autoencoders [263], self-supervised methods [99, 151, 456, 584], and deep generative models [156, 211, 262, 268, 340, 497, 556, 557]. Other important density estimation methods that have been applied to OOD detection include kernel density estimation [511], nearest neighbor methods [135], support vector machines [134, 585], and Gaussian mixture models [147]. However, these methods are not well suited for high-dimensional data such as images, text or audio, and have had little direct impact on the recent work on OOD detection.

Likelihood-based   Bishop [50] first proposed to use the likelihood assigned to data by a generative model as a measure for detecting anomalous data. Simply put, since the likelihood measures "how probable the model is to explain the data", OOD data is expected to give lower likelihoods than in-distribution data. Although, this method originally gave encouraging results, the advent of deep generative models and their application to high-dimensional data has lead many to observe likelihoods for OOD data that are higher than for in-distribution data [107, 254, 347, 473]. Such results have sparked interest in trying to explain this phenomenon and many works have proposed new scores for OOD detection derived from the likelihood.

In Ren et al. [552], the authors propose to use the likelihood ratio between a model trained on in-distribution data and a background model trained on perturbed in-distribution data as the OOD score. Serrà et al. [592] argue that the failure of deep generative models is due to the high-influence that the input complexity has on the likelihood. Therefore, they propose to use a general lossless image compression algorithm as a background model. Choi, Jang, and Alemi [107] propose to use the Watanabe information criterion (WAIC) computed from the likelihood [694, 695]. For variational autoencoders, other work proposes to refit the encoder on a test data example, hypothesizing that the likelihood of OOD data will improve more than for in-distribution data, and use this "likelihood regret" as the OOD score [716]. Maaløe et al. [434] provide initial results that a loosened variational bound on the likelihood, using only encoded representations from the top-most latent variables in a hierarchical variational autoencoder, can improve OOD detection performance. In Havtorn et al. [245], we show that variational autoencoders are surprisingly good at reconstructing OOD data and propose an improved score based on the likelihood ratio of such loosened bounds.

A different approach is to use the typicality set hypothesis [473]. The typicality set is the subset of the model full support in data space, where the model samples from, that does not overlap with regions of maximal likelihood. Nalisnick et al. [473] propose to use the typicality set as a test statistic for OOD detection

while Morningstar et al. [467] propose to use the related idea of density of states of the model. In Bergamin et al. [45] we use Fisher's method [184] to combine Rao's score test statistic [545] with the typicality set test statistic hence including information from both the gradient and the likelihood.

Reconstruction-based   A number of methods derive the OOD score from a reconstruction error. Among the first methods are Lyudchik [431] and Sakurada and Yairi [571] who note that dimensionality reduction helps separate inliers and outliers and propose to use deep autoencoders to reconstruct the input and evaluate the reconstruction error. Xia et al. [712] take a similar approach but also propose to inject discriminative information in the learning process. Drawing inspiration from Robust Principal Component Analysis, Zhou and Paffenroth [742] propose to first split the input data into a dense low-rank factor and a sparse factor, assuming that outliers are caught in the sparse factor, and then use a deep autoencoder to reconstruct the dense factor. Zong et al. [749] jointly learn a deep autoencoder and a Gaussian Mixture Model on the learned hidden representations and point out similarities of their method to neural variational inference [460]. Similar to the ensemble-based methods for supervised OOD detection, Chen et al. [95] propose to use an ensemble of autoencoders to reconstruct the input and use the median reconstruction error as the OOD score.

Although generative adversarial networks do not have the ability to encode a given data point, methods have been proposed to invert the generator to find an associated latent representation. This representation can then be used to reconstruct the input and a reconstruction error, or a discriminator score, can be used for OOD detection [397, 582].

Diffusion models have also been used for OOD detection via a reconstruction-based score. Graham et al. [215] add varying amounts of diffusion noise to an input image and show that reconstructions of OOD inputs from appropriate noise levels fall back onto the in-domain manifold resulting in high reconstruction error. Liu et al. [423] lift an image off its original manifold by sampling a number of masks, and then maps it towards the in-domain manifold with a diffusion model, using the median reconstruction error as the OOD score.

Representation-based   Denouden et al. [150] suggest that reconstruction-based approaches fail to capture particular anomalies that lie far from known inlier samples in latent space but near the latent dimension manifold defined by the parameters of the model. They propose to measure the Mahalanobis distance between the global Gaussian distribution of training set in latent space and an encoded test input. Xiao, Yan, and Amit [715] instead propose to use an existing, strong foundation model, pre-trained with a self-supervised objective, to extract features from the input, and then fit a Gaussian Mixture Model to the features

using the minimal Mahalanobis distance to the mixture components as the OOD score.

Several other works also use self-supervised representations of the in-domain data for OOD detection [47, 255]. Tack et al. [627] propose to use a contrastive objective to learn representations of the in-domain data contrasted with data augmented in-samples and use a softmax classifier trained on the representations to compute the OOD score. Sehwag, Chiang, and Mittal [590] present a similar approach but use the feature space Mahalanobis distance, similar to Denouden et al. [150]. Ahmadian and Lindsten [2] propose to use the latent representation of an invertible generative model to compute the OOD score.

3.3 VARIATIONAL AUTOENCODERS

Variational autoencoders (VAEs) belong to the class of deep generative models. These are models that use deep neural networks to approximate the underlying distribution of the unlabeled training data, from which they can then be used to generate new data samples. There exist at least six main classes of deep generative models: variational autoencoders [102, 340, 543, 557, 657], normalizing flow models [156, 157, 216, 339, 556], diffusion models [268, 612, 616, 658], energy-based models [162, 262, 378, 572], autoregressive models [496, 497, 540], and generative adversarial networks [11, 64, 211, 328].

Due to the assumptions they impose on the data modelling task and their probabilistic formulation, VAEs are interesting for representation learning as well as uncertainty quantification. To some degree, these properties set them apart from other deep generative models that focus more exclusively on data generation. In the following, we will introduce VAEs starting from the definition of latent variable models. We then derive the marginal likelihood objective bound used to train them, discuss its different components and approximation error, and highlight some central challenges of applying VAEs.

3.3.1 LATENT VARIABLE MODELS

The fundamental assumption underlying the definition of latent variable models is that a data point $\mathbf{x}$ is created via a generative process that involves one or more unobserved, stochastic latent variables $\mathbf{z}$. Latent variable model design centers around capturing this generative process and learning to generate new data points $\mathbf{x}$. The generative process is usually defined by a joint distribution $p(\mathbf{x}, \mathbf{z})$ over the data and latent variables. We can write the marginal distribution of the data as,

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z}) \, d\mathbf{z} \ . \tag{3.10}$$

The ability to infer the latent variables $\mathbf{z}$ from a given data point $\mathbf{x}$ is of interest in applications that focus on representation learning. Inference is performed by computing the posterior distribution $p(\mathbf{z}|\mathbf{x})$ via Bayes' theorem,

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})} \quad . \tag{3.11}$$

In many latent variable models, the joint distribution is factorized as $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$ including Gaussian mixture models [147], hidden Markov models [538], and probabilistic principal component analysis [639].

3.3.2 Variational inference

To model high-dimensional data, we want to use neural networks to represent the generative process. This leads to deep latent variable models which were pioneered by Kingma and Welling [340] and Rezende, Mohamed, and Wierstra [557]. The authors propose to factor the joint distribution and parameterize it using a neural network with parameters θ and let $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})$ where $p(\mathbf{z})$ is a prior.

Since the marginal and posterior distributions involve integrals over all possible values of the latent variable $\mathbf{z}$, using neural networks makes both intractable. This precludes training with commonly used methods such as expectation maximization and exact maximum likelihood estimation. Since Markov Chain Monte Carlo methods are usually too computationally expensive for large neural networks, this leaves variational inference as the scalable alternative [319].

Evidence lower bound By approximating the intractable posterior distribution $p(\mathbf{z}|\mathbf{x})$ with a variational distribution $q(\mathbf{z})$, we can derive a lower bound on the also intractable marginal log-likelihood, $\log p_\theta(\mathbf{x})$.

$$\log p_\theta(\mathbf{x}) = \log \int p_\theta(\mathbf{x}, \mathbf{z}), d\mathbf{z}$$

$$= \log \int q(\mathbf{z}) \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \, d\mathbf{z} \tag{3.12}$$

$$\geqslant \int q(\mathbf{z}) \log \left(\frac{p_\theta(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \right) d\mathbf{z}$$

$$= \mathbb{E}_{q(\mathbf{z})} \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \right] \equiv \mathcal{L}(\mathbf{x}; \theta, q) \quad . \tag{3.13}$$

Since we can evaluate the generative model $p_\theta(\mathbf{x}, \mathbf{z})$ and are free to define $q(\mathbf{z})$ as we please, this bound can be made tractable.

The usual approach in variational inference is to fit an approximate posterior q(**z**) per data point. However, for large datasets, as is common in natural language processing, this can be too computationally expensive. Instead, VAEs amortize the cost of variational inference by parameterizing the approximate posterior with a neural network that transforms any data point **x** into the parameters of the corresponding conditional distribution. We denote this by $q_\phi(\mathbf{z}|\mathbf{x})$ where ϕ are the parameters of this inference model. This makes it possible to efficiently infer the approximate posterior distribution for any input **x**. Using amortized variational inference, the ELBO is written as,

$$\mathcal{L}(\mathbf{x}; \theta, \phi) \equiv \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] \leqslant \log p_\theta(\mathbf{x}) \ . \tag{3.14}$$

Although analytically evaluating the ELBO is intractable due to the expectation, it can be approximated by sampling and then used to train VAEs. We will return to this after examining the ELBO in more detail.

KL-divergence to prior   Since VAEs factorize the joint distribution, we can also write the ELBO in (3.14) as,

$$\mathcal{L}(\mathbf{x}; \theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x}|\mathbf{z}) \right] + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \frac{p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right]$$

$$= \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x}|\mathbf{z}) \right]}_{\text{reconstruction loss}} - \underbrace{D_{\mathrm{KL}} \left(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z}) \right)}_{\text{KL-divergence to prior}} \ . \tag{3.15}$$

This reveals a commonly used interpretation of the ELBO: The expected log-likelihood of the generative model $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x}|\mathbf{z}) \right]$ acts as a reconstruction loss; since mapping many **z** to a single **x** is hard, it encourages the model to make $q_\phi(\mathbf{z}|\mathbf{x})$ as peaky as possible. The negative KL-divergence between the approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$ and the prior $p(\mathbf{z})$ acts as a regularizer that forces the approximate posterior to be within the support of the prior (reverse KL-divergence). This helps prevent overfitting but importantly also enables ancestral sampling from the prior to generate high quality data.

By maximizing $\mathcal{L}(\mathbf{x}; \theta, \phi)$ we must minimize the KL-divergence. However, an optimal KL-divergence of zero requires $q_\phi(\mathbf{z}|\mathbf{x}) = p(\mathbf{z})$ which would mean that **z** has become independent of **x**. This state is a poor local optimum and is usually referred to as *posterior collapse*. It is particularly prone to happen for strong generative models for instance with autoregressive dependencies $p(\mathbf{x}_t|\mathbf{x}_{1:t}, \mathbf{z})$. Some works have proposed to mitigate posterior collapse by tempering the KL-divergence term [7, 259] or by adding additional terms to the ELBO [741]. A well-fitted VAE will therefore have maximized the expected log-likelihood of the

data under the generative model $p_\theta(\mathbf{x}|\mathbf{z})$ by learning an informative latent variable and thus have a nonzero KL-divergence $D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})\right)$.

Inference gap   The ELBO is a lower bound on the marginal likelihood, so it is interesting to examine when the bound holds exactly. As is standard for variational Bayesian methods, we can equivalently take the objective of the VAE to be the minimization of the KL-divergence of the approximate posterior to the true posterior, $D_{KL}\left(p_\theta(\mathbf{z}|\mathbf{x}) \| q_\phi(\mathbf{z}|\mathbf{x})\right)$. We can then rewrite this objective as follows.

$$
\begin{aligned}
D_{KL}\left(p_\theta(\mathbf{z}|\mathbf{x}) \| q_\phi(\mathbf{z}|\mathbf{x})\right) &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{z},\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})p_\theta(\mathbf{x})}\right] \\
&= \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{z},\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\right]}_{\text{ELBO}} - \underbrace{\log p_\theta(\mathbf{x})}_{\text{marginal likelihood}} .
\end{aligned}
\tag{3.16}
$$

We can see that the difference between the ELBO and the marginal likelihood is exactly equal to the KL-divergence of the approximate posterior to the true posterior $D_{KL}\left(p_\theta(\mathbf{z}|\mathbf{x}) \| q_\phi(\mathbf{z}|\mathbf{x})\right)$. This difference is called the *inference gap*. This also means that the approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$ that maximizes the ELBO also best approximates the true posterior, as measured by KL-divergence.

   We can analyze the inference gap further [136]. By amortizing the variational inference we introduce a gap compared to the alternative of fitting a posterior per data point. This is referred to as the *amortization gap* and can be reduced by using large capacity amortized posteriors. In case the true posterior does not belong to the chosen class of variational posteriors, the inference gap cannot be reduced to zero even with infinite capacity. The remaining gap is referred to as the *approximation gap*. The inference gap is the sum of these two contributions. When the inference gap is closed, the ELBO becomes equal to the marginal likelihood, the value of which then depends on how well the generative model is fitted [136, 189].

3.3.3   Maximum likelihood estimation

VAEs are almost always defined as mixed-effect models: VAEs use variational Bayes to infer a posterior distribution over latent variables, but their parameters are estimated using maximum likelihood. Hence, parameter estimation entails jointly optimizing the generative model $p_\theta(\mathbf{x}|\mathbf{z})$ and the variational approximation $q_\phi(\mathbf{z}|\mathbf{x})$ with respect to θ and ϕ to find point estimates. Gradient-based op-

timization works well for neural networks, but the large amounts of data force the use of mini-batch, or stochastic, gradient descent. Furthermore, we must approximate the expectation over the approximate posterior in the ELBO (3.14) to compute its gradients w.r.t. ϕ and θ. We will look into how batching affects the form of the ELBO in section 3.3.5 and move on with posterior sampling and gradient estimation here.

Sampling   Since we have freedom to select the approximate posterior, we can choose it such that it is easy to sample and then use that property to form a Monte Carlo estimate of the ELBO, $\mathcal{L}(\mathbf{x}; \theta, \phi)$,

$$\widehat{\mathcal{L}}_S(\mathbf{x}; \theta, \phi) = \frac{1}{S} \sum_{s=1}^{S} \log \frac{p_\theta(\mathbf{x}|\mathbf{z}_s)p(\mathbf{z}_s)}{q_\phi(\mathbf{z}_s|\mathbf{x})} \quad , \tag{3.17}$$

where $\mathbf{z}_s \sim q_\phi(\mathbf{z}|\mathbf{x})$ are samples from the approximate posterior. For some choices of prior and approximate posterior, such as Gaussians, the KL-divergence term of the ELBO can be computed analytically which reduces estimator variance. Such estimators can be written as,

$$\widehat{\mathcal{L}}_S(\mathbf{x}; \theta, \phi) \equiv \frac{1}{S} \sum_{s=1}^{S} \log p_\theta(\mathbf{x}|\mathbf{z}_s) - D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}) \,\|\, p(\mathbf{z})\right) \quad . \tag{3.18}$$

Since $\widehat{\mathcal{L}}_S(\mathbf{x}; \theta, \phi)$ is the sample average, it is an unbiased estimator of the ELBO, $\mathbb{E}\left[\widehat{\mathcal{L}}_S(\mathbf{x}; \theta, \phi)\right] = \mathcal{L}_S(\mathbf{x}; \theta, \phi)$, and hence a biased estimator of the marginal likelihood, $\log_\theta p(\mathbf{x})$.

   In the above we arrived at $\widehat{\mathcal{L}}_S(\mathbf{x}; \theta, \phi)$ by first deriving the ELBO of (3.13) as a lower bound on the marginal likelihood and then approximating that bound by Monte Carlo sampling. Instead, Burda, Grosse, and Salakhutdinov [69] propose to Monte Carlo sample the expectation inside the logarithm of (3.12) directly, without first deriving the ELBO. They obtain

$$\widehat{\mathcal{L}}_S^{IWAE}(\mathbf{x}; \theta, \phi) \equiv \log\left(\frac{1}{S} \sum_{s=1}^{S} \frac{p_\theta(\mathbf{x}|\mathbf{z}_s)p(\mathbf{z}_s)}{q_\phi(\mathbf{z}_s|\mathbf{x})}\right) = \frac{1}{S} \sum_{s=1}^{S} \log w_\theta(\mathbf{x}, \mathbf{z}_s) \quad , \tag{3.19}$$

where $w_\theta(\mathbf{x}, \mathbf{z}_s)$ can be seen as importance weights. Note that the location of the logarithm is the only difference with the estimator in (3.17). By taking the expectation of this estimator, we can see that it too is a lower bound on the marginal

likelihood,

$$\mathcal{L}_S^{\text{IWAE}}(\mathbf{x}; \theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\widehat{\mathcal{L}}_S^{\text{IWAE}}(\mathbf{x}; \theta, \phi)\right]$$

$$= \mathbb{E}_{\mathbf{z}_s \sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\frac{1}{S}\sum_{s=1}^{S}\log\left(w_\theta(\mathbf{x}, \mathbf{z}_s)\right)\right]$$

$$\leqslant \log \mathbb{E}_{\mathbf{z}_s \sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\frac{1}{S}\sum_{s=1}^{S}w_\theta(\mathbf{x}, \mathbf{z}_s)\right]$$

$$= \log p_\theta(\mathbf{x}) \ . \tag{3.20}$$

From this and the law of large numbers, it can be shown that for all S,

$$\log p_\theta(\mathbf{x}) \geqslant \mathcal{L}_{S+1}^{\text{IWAE}}(\mathbf{x}; \theta, \phi) \geqslant \mathcal{L}_S^{\text{IWAE}}(\mathbf{x}; \theta, \phi) \ , \tag{3.21}$$

and $\lim_{S \to \infty} \mathcal{L}_S^{\text{IWAE}}(\mathbf{x}; \theta, \phi) \to \log p_\theta(\mathbf{x})$, if $w_\theta(\mathbf{x}, \mathbf{z}_s)$ are bounded. Hence, $\widehat{\mathcal{L}}_S^{\text{IWAE}}$ is a biased estimator of $\log p_\theta(\mathbf{x})$ but with a bias that goes to zero as the number of samples goes to infinity. With $S = 1$, the estimator is equivalent to the regular ELBO (3.14) and so, it is always as tight, or tighter, than the ELBO [69].

Reparameterization   Computing the gradient of the ELBO (3.14) with respect to θ is straightforward,

$$\nabla_\theta \mathcal{L}(\mathbf{x}; \theta, \phi) = \nabla_\theta \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\nabla_\theta \log p_\theta(\mathbf{x}, \mathbf{z})\right] \ . \tag{3.22}$$

However, the gradient with respect to the ϕ is more challenging. Since ϕ occurs in the expectation itself, we may not move the gradient inside the expectation. However, by reparameterizing the latent variable $\mathbf{z}$ as a deterministic function of the input $\mathbf{x}$ and a random variable $\epsilon \sim p(\epsilon)$, $\mathbf{z} = g_\phi(\mathbf{x}, \epsilon)$, we can write the gradient as a path-wise derivative [340, 462, 557],

$$\nabla_\phi \mathcal{L}(\mathbf{x}; \theta, \phi) = \nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right]$$

$$= \mathbb{E}_{p(\epsilon)}\left[\nabla_\phi \log \frac{p_\theta\left(\mathbf{x}, g_\phi(\mathbf{x}, \epsilon)\right)}{q_\phi\left(g_\phi(\mathbf{x}, \epsilon)|\mathbf{x}\right)}\right] \ . \tag{3.23}$$

The most important instantiation of this reparameterization is the diagonal Gaussian case. By choosing $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mu_\phi(\mathbf{x}), \mathbf{I}\sigma_\phi(\mathbf{x}))$ it is possible to sample differentiably via the reparameterization trick by setting $g_\phi(\mathbf{x}, \epsilon) = \mu_\phi(\mathbf{x}) + \sigma_\phi(\mathbf{x}) \odot \epsilon$ with $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. By assuming the covariance matrix is diagonal, this becomes

a mean-field approximation which imposes independence between the dimensions of $\mathbf{z}$. The gradient for $\mathcal{L}_S^{\text{IWAE}}$ is derived similarly.

Although the variance of the path-wise derivative is lower than most alternatives, such as score function estimators, it is not negligible, and a number of works have attempted to reduce it. Roeder, Wu, and Duvenaud [562] note that a score function estimator can be factored out of the path-wise derivative and propose to ignore it to reduce gradient variance. Rainforth et al. [542] show that this problem is exacerbated for the importance weighted ELBO [69] and that increasing the number of samples in the importance weighted bound also increases gradient variance for the inference network, hurting its ability to learn useful representations. After showing that removing the score function factor introduces bias, Tucker et al. [651] propose to reparameterize it too, giving rise to an unbiased, low variance gradient estimator that improves with more samples. Later work has generalized this estimator to hierarchical models [35].

3.3.4 HIERARCHICAL MODELS

The mean-field assumption imposed by $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{I}\sigma^2)$ can often be more restrictive for model expressiveness than we would like. While learning a full covariance is a simple solution, it is not always sufficiently computationally efficient and, in any case, enables learning only linear covariance between elements of $\mathbf{z}$. The drawbacks of mean-field approximation and linear covariance have lead to research into learning hierarchies of several non-linearly dependent latent variables; an idea well in line with the usual motivations behind deep neural networks such as efficient, compositional representation [377]. Such hierarchical models are usually formalized by introducing a set of L latent variables $\mathbf{z} = \mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(L)}$ and letting the generative model be defined as,

$$p_\theta(\mathbf{x}|\mathbf{z}) = p(\mathbf{x}|\mathbf{z}^{(1)})p_\theta(\mathbf{z}^{(1)}|\mathbf{z}^{(2)}) \cdots p_\theta(\mathbf{z}^{(L-1)}|\mathbf{z}^{(L)})p(\mathbf{z}^{(L)}) \ . \qquad (3.24)$$

This top-down generative model can be efficiently sampled via ancestral sampling; first $\mathbf{z}^{(L)}$ is drawn from the prior $p(\mathbf{z}^{(L)})$, and then each $\mathbf{z}^{(l)}$ is drawn from the corresponding $p(\mathbf{z}^{(l)}|\mathbf{z}^{(l+1)})$ until we can draw $p(\mathbf{x}|\mathbf{z}^{(1)})$.

The inference model can then be defined in two ways respectively referred to as *bottom-up* [69]

$$q_\phi(\mathbf{z}|\mathbf{x}) = q_\phi(\mathbf{z}^{(1)}|\mathbf{x}) \prod_{i=2}^{L} q_\phi(\mathbf{z}^{(i)}|\mathbf{z}^{(i-1)}) \qquad (3.25)$$

and *top-down* [614]

$$q_\phi(\mathbf{z}|\mathbf{x}) = q_\phi(\mathbf{z}^{(L)}|\mathbf{x}) \prod_{i=1}^{L-1} q_\phi(\mathbf{z}^{(i)}|\mathbf{z}^{(i+1)}) \ . \qquad (3.26)$$

A variant of the variational autoencoder that employs a bidirectional inference network has also been proposed [434]. Regardless of the choice of inference model, the resulting hierarchical VAE is trained using the ELBO (3.14) and the reparameterization trick. Recent architectural advances have alleviated the posterior collapse problem and made it possible to train VAEs with many latent variables [102, 434, 657]. A key modelling choice in these works is conditioning each latent variable directly on the input via residual connections.

3.3.5 Mutual information interpretation

Training VAEs by stochastic gradient descent involves first sampling a mini-batch of data points $\mathbf{x}$ from the empirical distribution of training data $\hat{p}(\mathbf{x})$ and then evaluating the gradient of the ELBO with respect to the parameters θ and ϕ. We can note that evaluating the ELBO with a mini-batch sampled from the training data corresponds to taking an expectation over the data distribution $\hat{p}(\mathbf{x})$.

To gain some additional insight into the objective that we are optimizing, we will use this expectation over $\hat{p}(\mathbf{x})$ to form a factorization similar to (3.15) but with the aggregated posterior $q_\phi(\mathbf{z}) = \int q_\phi(\mathbf{z}|\mathbf{x})\hat{p}(\mathbf{x})\,d\mathbf{x}$ in the KL-divergence [645]. Under the expectation over the data distribution, the marginal likelihood is lower bounded by that expectation over the ELBO,

$$\mathbb{E}_{\hat{p}(\mathbf{x})}\left[\log p_\theta(\mathbf{x})\right] \geqslant \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \ , \qquad (3.27)$$

where we have defined $q_\phi(\mathbf{x},\mathbf{z}) = q_\phi(\mathbf{z}|\mathbf{x})\hat{p}(\mathbf{x})$ for ease of notation. This is simply (3.14) under the extra expectation. We leave the reconstruction loss in (3.15) as is and focus on the expectation over the KL-divergence to the prior. We rewrite it as follows,

$$
\begin{aligned}
\mathbb{E}_{\hat{p}(\mathbf{x})}\left[D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})\,\|\,p(\mathbf{z})\right)\right] &= \mathbb{E}_{q_\phi(\mathbf{z},\mathbf{x})}\left[\log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}\right] \\
&= \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log \frac{1}{p(\mathbf{z})}\right] + \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{x},\mathbf{z})}{q(\mathbf{x})}\right] \\
&= \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{z})}{p(\mathbf{z})}\right] + \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{x},\mathbf{z})}{q(\mathbf{x})q_\phi(\mathbf{z})}\right] \\
&= D_{KL}\left(q_\phi(\mathbf{z})\,\|\,p(\mathbf{z})\right) + I_{q_\phi(\mathbf{x},\mathbf{z})}[\mathbf{x};\mathbf{z}] \ . \qquad (3.28)
\end{aligned}
$$

Inserting this back into the ELBO in (3.15), we get,

$$\mathbb{E}_{\hat{p}(\mathbf{x})}\left[\mathcal{L}(\mathbf{x};\theta,\phi)\right] = \underbrace{\mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]}_{\text{average reconstruction}} - \underbrace{D_{KL}\left[q_\phi(\mathbf{z})\,\|\,p(\mathbf{z})\right]}_{\text{marginal KL to prior}} - \underbrace{I_{q_\phi(\mathbf{x},\mathbf{z})}\left[\mathbf{x};\mathbf{z}\right]}_{\text{mutual information}} \ .$$

$$(3.29)$$

In this form, the KL-divergence term of (3.15) has been factored into the marginal KL-divergence of the aggregated posterior to the prior and the mutual information between the data and the latent variables. The marginal KL-divergence is minimized by making the aggregated posterior $q_\phi(\mathbf{z})$ match the prior $p(\mathbf{z})$. Contrary to the KL-divergence term of (3.15), driving the marginal KL-divergence to zero does not enforce independence between $\mathbf{x}$ and $\mathbf{z}$. However, this alternative form reveals that the ELBO is maximized by minimizing the mutual information between $\mathbf{x}$ and $\mathbf{z}$, $I_{q_\phi(\mathbf{x},\mathbf{z})}[\mathbf{x};\mathbf{z}]$. This result indicates that the usual interpretation of $\mathbf{z}$ as a representation of $\mathbf{x}$ is not a fundamental property arising from the ELBO, or from the assumptions from which the VAE was derived. This challenges the usefulness of variational autoencoders as representation learners.

Applications of VAEs   Despite the challenges facing the training of VAEs, they have been successfully applied to a number of tasks including image generation [340, 557], image inpainting [518], image super-resolution [103, 613], and speech synthesis [278, 279]. VAEs have also proven themselves useful for semi-supervised learning [343, 344, 434].

Self-supervised learning   An alternative approach to representation learning is provided by self-supervised learning [99, 151, 456, 584] which is a form of unsupervised learning where the training objective is derived from the data itself. While it is fair to say that not all self-supervised training objectives have as principled a motivation as the VAE ELBO does, they have shown impressive results within the fields of natural language processing [99, 151], speech processing [584], and computer vision [99]. We provide a review and comparison of self-supervised methods and variational autoencoders for speech representation learning in chapter 6. In appendix A we provide a comprehensive introduction to and review of self-supervised learning in speech recognition.

PART II

UNSUPERVISED UNCERTAINTY ESTIMATION

HIERARCHICAL VAEs KNOW WHAT THEY DON'T KNOW

This chapter is a piece of original research by the candidate, previously published else-where and reprinted here with permission:

[A] **Havtorn, J. D.**, Frellsen, J., Hauberg, S., Maaløe, L., "Hierarchical VAEs Know What They Don't Know". In: *Proceedings of the 38th International Conference on Machine Learning (ICML)*. Virtual: PMLR, 2021. arXiv: 2102.08248 [main author] [245]

ABSTRACT

Deep generative models have been demonstrated as state-of-the-art density estimators. Yet, recent work has found that they often assign a higher likelihood to data from outside the training distribution. This seemingly paradoxical behavior has caused concerns over the quality of the attained density estimates. In the context of hierarchical variational autoencoders, we provide evidence to explain this behavior by out-of-distribution data having in-distribution, low-level features. We argue that this is both expected and desirable behavior. With this insight in hand, we develop a fast, scalable, and fully unsupervised likelihood-ratio score for OOD detection that requires data to be in-distribution across all feature-levels. We benchmark the method on a vast set of data and model combinations and achieve state-of-the-art results on out-of-distribution detection.

4.1 INTRODUCTION

The reliability and safety of machine learning systems applied in the real-world is contingent on the ability to detect when an input is different from the training distribution. Supervised classifiers built as deep neural networks are well-known to misclassify such *out-of-distribution* (OOD) inputs to known classes with high confidence [212, 485]. Several approaches have been suggested to equip deep classifiers with OOD detection capabilities [152, 253, 254, 368]. But, such methods are inherently supervised and require in-distribution labels or examples of OOD data limiting their applicability and generality.

Unsupervised generative models that estimate an explicit likelihood should understand what it means to be in- and out-of-distribution without requiring labels or examples of OOD data. By directly modeling the training distribution,
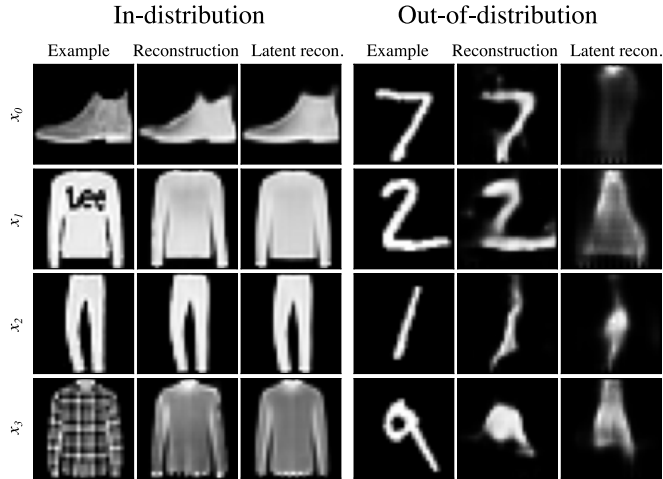
Figure 4.1: Reconstructions using a hierarchical VAE trained on FashionMNIST. Reconstruction quality of OOD data is comparable to in-distribution data, resulting in high likelihoods and poor OOD discrimination. By sampling the k bottommost latent variables from the conditional prior distribution $p_\theta(\mathbf{z}^{(\geq l)}|\mathbf{z}^{(>l)})$ (latent reconstructions) instead of the approximate posterior $q_\phi(\mathbf{z}^{(>l)}|\mathbf{z}^{(<l)})$, the model reconstructs from the training distribution resulting in lower $p_\theta(\mathbf{x}|\mathbf{z})$ for OOD data.

such models are expected to assign low likelihoods to OOD data as it originates from regions of little or no support under the learned density [50]. Recent advances in deep generative models [339, 340, 498, 557, 573] have enabled learning high quality generative models on complex data such as natural images, sequences including audio [496] and graphs [346]. However, recent observations have brought into question the quality of the learned density estimates by showing that they often assign higher likelihoods to OOD data than to in-distribution data [107, 472]. Many complex data distributions can be explained to a large degree by low-level features, e.g. edges in images. However, such features do not explain high-level semantics of the data and may inhibit OOD detection [472, 552]

In this paper, we examine the failure cases of deep generative models on OOD detection tasks within the context of hierarchical VAEs, and make the following contributions:

(i) We provide evidence that the root cause of OOD failures is that learned low-level features generalize well across datasets and dominate the estimated likelihoods.

(ii) We then propose a fast, scalable, and fully unsupervised likelihood-ratio score for OOD detection that is explicitly developed to ensure that data should be in-distribution across all feature levels, which prevents the low-level features from dominating.

(iii) With the likelihood-ratio score, we demonstrate state-of-the-art performance across a wide range of known OOD failure cases.

4.2 WHY DOES OOD DETECTION FAIL?
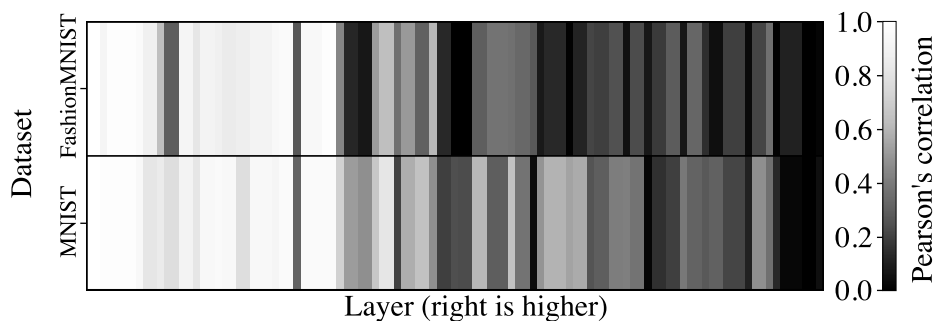
Figure 4.2: Absolute correlations between data representations in all layers of the inference network of a hierarchical VAE trained on FashionMNIST and of another trained on MNIST. We compute the correlation between the representations of the two different models given the same data, FashionMNIST (top) and MNIST (bottom).

The inability to detect out-of-distribution data with deep generative models is surprising. Before the advent of deep generative models, this was not considered a major issue for probabilistic models [50]. Is the failure due to model pathologies or something different?

Deep learning models are generally believed to form hierarchies of representations that range from low-level features to more conceptual ones related to semantics [41]. This has also been observed within deep generative models [102, 434]. For image data there is a trend that the low-level features are quite similar across models (edge detectors, etc.). This raises the question to what extent such features are relevant when detecting OOD data, also suggested by [472] and examined for Glow and PixelCNN in [581]. To investigate, we train two hierarchical VAEs (section 4.3.2) on FashionMNIST and MNIST, respectively, and compute the between-models correlation of the extracted features of in-distribution

Example   Reconstructions from latent hierarchy (right is higher)
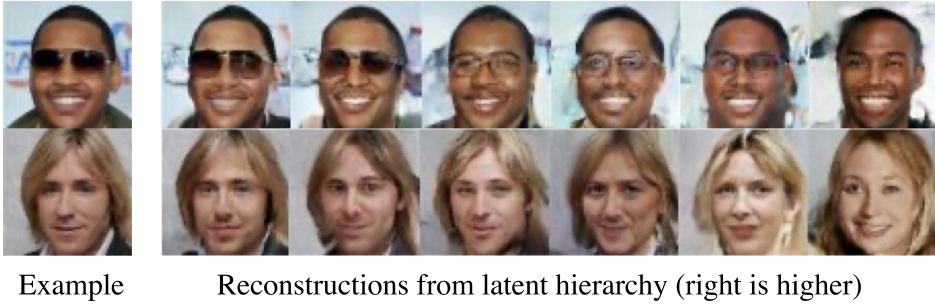
Figure 4.3: Reconstructions of in-distribution data (CelebA) of the BIVA model using higher latent variables [434]. The higher the latent variable, the more the reconstructions fall into the mode of the learned distribution. It is more common to wear regular glasses than sunglasses but most common not to wear glasses at all. A man with long hair collapses into the mode of the more common long-haired woman.

data and OOD data. The result appears in figure 4.2. We observe that features extracted in the early layers (low-level features) correlate strongly between the two models, and that this correlation drops as we get into later layers. This suggests that low-level features do not carry much information for OOD detection.

To shed further light on the impact of semantic versus low-level features, we look at model reconstructions of images with a hierarchical VAE (figure 4.3). To study the feature hierarchy, we replace the inference distribution with the corresponding conditional prior in the first layers of the model to see what information is lost. We observe that as more layers rely on the prior, more details are lost. Sunglasses, which are uncommon, are first replaced by more common glasses, and then finally disappear. This suggests that as we fall back to the conditional priors of each layer, we are pushed closer to local modes of the modeled distribution.

Finally, we look at reconstructions of out-of-distribution data. figure 4.1 illustrates that MNIST data is surprisingly well reconstructed by a hierarchical VAE trained on FashionMNIST. Similar results have been found elsewhere [716]. We repeat the previous experiment and replace inference distributions by their corresponding conditional prior, and now observe that reconstructions from higher latent layers become increasingly similar to the data on which the model was trained. The reliance on conditional priors seems to prevent accurate reconstruction of out-of-distribution data. Some details are lost on in-distribution data too, but the distinction between that and out-of-distribution data becomes more clear.

These observations lead to our main hypothesis. The lowest latent variables in a hierarchical VAE learn generic features that can describe a wide range of data. This enables the model to achieve high rates of compression and high likelihoods, even on out-of-distribution data as long as the learned low-level features are appropriate. We further suggest that OOD data are in-distribution with respect to these low-level features, but not with respect to semantic ones.

4.3 BACKGROUND AND RELATED WORK

4.3.1 VARIATIONAL AUTOENCODERS

The variational autoencoder (VAE) [340, 557] is a framework for constructing deep generative models defined by an observed variable $\mathbf{x}$ and a stochastic latent variable $\mathbf{z}$. Typically, a neural network with parameters θ is chosen to parameterize the generative distribution $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})$, where the prior $p(\mathbf{z})$ is commonly a standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The true posterior $p(\mathbf{z}|\mathbf{x})$ is generally not analytically tractable and is approximated by a variational distribution $q_\phi(\mathbf{z}|\mathbf{x})$ parameterized via another neural network with parameters ϕ. The approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$ is most often a diagonal covariance Gaussian. The model parameters θ and variational parameters ϕ are jointly optimized by maximizing the *evidence lower bound* (ELBO),

$$\log p_\theta(\mathbf{x}) \geqslant \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] \equiv \mathcal{L}(\mathbf{x}; \theta, \phi) \ . \tag{4.1}$$

For brevity, we will denote $\mathcal{L}(\mathbf{x}; \theta, \phi)$ as $\mathcal{L}(\mathbf{x})$ or $\mathcal{L}$. The reparameterization trick is used to backpropagate gradients through the stochastic latent variables with low variance.

The VAE is defined with a single latent variable which limits the ability to learn a high likelihood representation of complex input distributions, e.g. natural images. There exists a few complementary approaches to make the VAE more flexible: (i) model a more expressive variational distribution $q_\phi(\mathbf{z}|\mathbf{x})$ or prior distribution $p_\theta(\mathbf{z})$ [344, 556], (ii) model a more expressive posterior distribution $p_\theta(\mathbf{x}|\mathbf{z})$ e.g. with an autoregressive decoder [497] and (iii) learn a deeper hierarchy of latent variables [69, 614]. Here, we focus on the latter.

4.3.2 HIERARCHICAL VARIATIONAL AUTOENCODERS

Hierarchical VAEs are a family of probabilistic latent variable models which extends the basic VAE by introducing a hierarchy of L latent variables $\mathbf{z} = \mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(L)}$. The most common generative model is defined from the top down as $p_\theta(\mathbf{x}|\mathbf{z}) =$

$p_\theta(\mathbf{x}|\mathbf{z}^{(1)})p_\theta(\mathbf{z}^{(1)}|\mathbf{z}^{(2)}) \cdots p_\theta(\mathbf{z}^{(L-1)}|\mathbf{z}^{(L)})$. The inference model can then be defined in two ways respectively referred to as *bottom-up* [69]

$$q_\phi(\mathbf{z}|\mathbf{x}) = q_\phi(\mathbf{z}^{(1)}|\mathbf{x}) \prod_{i=2}^{L} q_\phi(\mathbf{z}^{(i)}|\mathbf{z}^{(i-1)}) \tag{4.2}$$

and *top-down* [614]

$$q_\phi(\mathbf{z}|\mathbf{x}) = q_\phi(\mathbf{z}^{(L)}|\mathbf{x}) \prod_{i=L-1}^{1} q_\phi(\mathbf{z}^{(i)}|\mathbf{z}^{(i+1)}) \ . \tag{4.3}$$

Regardless of the choice of inference model, a hierarchical VAE is still trained using the ELBO (4.1).

Until recently, hierarchical VAEs gave inferior likelihoods compared to state-of-the-art autoregressive [267] and flow-based models [573]. This was changed by Maaløe et al. [434], Vahdat and Kautz [657], and Child [102], which introduced complementary methods to extend the number of latent variables to a very deep hierarchy resulting in state-of-the-art likelihood performance.

In this paper we employ a simple hierarchical VAE with bottom-up inference paths and the more powerful BIVA variant with a bidirectional (top-down and bottom-up) inference model [434]. We employ skip connections between latent variables but omit them for brevity.

4.3.3 OUT-OF-DISTRIBUTION DETECTION

So far, no reliable direct likelihood-based method has been found for fully un-supervised deep generative model OOD detection. A major line of work considers developing new scores that are more reliable than the likelihood. This includes the *typicality* test presented by Nalisnick et al. [473] which is an OOD detection test based on the typicality of a batch of potentially OOD examples. This approach however requires a batch of examples from the same class (OOD or not) which limits its practical applicability. In Ren et al. [552], the *likelihood ratio* between a primary model and a background model was shown to be an effective score for OOD detection. However, to train the background model, the in-distribution data is perturbed via a data augmentation technique that is designed with knowledge about the confounding factors between the in-distribution data and the OOD data. Furthermore, it is tuned towards high performance on a known OOD dataset. Serrà et al. [592] take a similar approach and attribute the failure to detect OOD data to the high influence of the input complexity on the likelihood and choose a generic lossless compression algorithm as the background model. Although this method gives good results, no single best choice of compression algorithm exists for all types of OOD data, and any particular choice encodes prior knowledge about the data into the detection method. Both these methods can be seen as correcting for low-level features of the OOD data

being assigned high model likelihood by using a second model focused exclusively on these features.

Similar to these methods, the majority of the approaches to OOD detection make assumptions about the nature of the OOD data. The assumptions encompass using labels on the in-distribution data [5, 253, 368, 389, 400], examples of OOD data [254], augmenting in-distribution data to mimic it [552], or assuming a certain data type [592]. Any of these assumptions encode implicit biases into the model about the attributes of OOD data which, in turn, might impair performance on truly unknown data examples (unknown unknowns).

While some of these methods achieve very good results on OOD detection with autoregressive models [498, 573] and invertible flow-based models [339], it was recently shown that they can be much less effective for VAEs [716] highlighting the need for a more reliable OOD score for VAEs. Although VAEs have the same failure cases as autoregressive and flow-based models, the caveat is that the difference in the likelihood is generally not as big and reconstructions of OOD can be surprisingly good [716]. Xiao, Yan, and Amit [716] alleviate this by refitting the inference network, as previously proposed by Cremer, Li, and Duvenaud [136] and Mattei and Frellsen [448], to a potentially OOD example and measuring the so-called *likelihood regret*. However, refitting the inference network can be computationally expensive, especially for the large hierarchical VAEs that are used to model complex data [102, 434, 657]. Furthermore, this scales poorly to large amounts of potentially OOD examples as the optimization is done per example.

A few methods have approached OOD detection in a completely unsupervised fashion [107, 434, 716]. The work of Maaløe et al. [434] is the most related to ours. They introduce BIVA, a deep hierarchy of stochastic latent variables with a top-down and bottom-up inference model and achieve state-of-the-art likelihood scores. They also provide early results indicative that a looser likelihood bound may have value in OOD detection. In this paper, we provide an explanation of those results, and significantly improve upon them.

4.4 OOD DETECTION WITH HIERARCHICAL VAEs

4.4.1 A BOUND FOR SEMANTIC OOD DETECTION

If the lowest latent variable in the VAE hierarchy codes for a large part of the low-level features required to reconstruct the input with high accuracy, as exemplified in figures 4.1 to 4.3, then $p_\theta(\mathbf{x}|\mathbf{z}^{(1)})$ will be high for both in- and out-of-distribution data. Hence, any OOD detection capabilities based on the ELBO $\mathcal{L} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z}^{(1)})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z}))$ from (4.1) relies on the KL-term for OOD detection. For a bottom-up hierarchical VAE, the KL-term $D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel$

$p(\mathbf{z})$) can be expressed by a hierarchical sum,

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\sum_{i=1}^{L-1} \log \frac{p_\theta(\mathbf{z}^{(i)}|\mathbf{z}^{(i+1)})}{q_\phi(\mathbf{z}^{(i)}|\mathbf{z}^{(i-1)})} + \log \frac{p_\theta(\mathbf{z}^{(L)})}{q_\phi(\mathbf{z}^{(L)}|\mathbf{z}^{(L-1)})}\right] \ . \tag{4.4}$$

In general, the absolute log-ratios grow with $\dim(\mathbf{z}^{(i)})$ as the individual log probability terms are computed by summing over the dimensionality of $\mathbf{z}^{(i)}$. This means that the value of the KL-term is dominated by terms where $\mathbf{z}^{(i)}$ is high-dimensional. We refer to appendix B.3 for a more detailed argument. Since hierarchical VAEs are generally constructed with a bottleneck type structure, the terms corresponding to latent variables towards the top of the hierarchy will have a vanishing influence on the value of the KL-term. However, as the semantic information most relevant for OOD detection has a tendency to be represented in the top-most latent variables, this makes OOD detection using the regular ELBO difficult, even for state-of-the-art models. This behavior has also been reported by Xiao, Yan, and Amit [716].

To shift the ELBO from primarily being based on the approximate posterior of the lowest latent variables to instead focus on the conditional prior, Maaløe et al. [434] introduced slightly different likelihood lower bound defined as

$$\mathcal{L}^{>k} = \mathbb{E}_{p_\theta(\mathbf{z}^{(\leq k)}|\mathbf{z}^{(>k)})q_\phi(\mathbf{z}^{(>k)}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z}^{(>k)})}{q_\phi(\mathbf{z}^{(>k)}|\mathbf{x})}\right] \tag{4.5}$$

where $k \in \{0, 1, \dots, L\}$ (see appendix B.4 for the derivation). We note that $\mathcal{L}^{>0}$ is the regular ELBO ((4.1)) and that empirically we always observe that $\mathcal{L} \geq \mathcal{L}^{>k} \,\forall\, k$, although this need not hold in general. The core idea behind this variation on the ELBO is to sample the k lowest latent variables from the conditional prior $\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(k)} \sim p_\theta(\mathbf{z}^{(\leq k)}|\mathbf{z}^{(>k)})$ and only the $L - k$ highest from the approximate posterior $\mathbf{z}^{(k+1)}, \dots, \mathbf{z}^{(L)} \sim q_\phi(\mathbf{z}^{(>k)}|\mathbf{x})$. Importantly, this has the effect that the data likelihood $p(\mathbf{x}|\mathbf{z})$ is dependent on the approximate posterior through a latent variable $\mathbf{z}^{(k+1)}$ different from $\mathbf{z}^{(1)}$ for all $k \geq 1$. Thereby, the likelihood can be evaluated with a reconstruction from each of the latent variables $\mathbf{z}^{(k)}$ of the hierarchical VAE. Hence, we can now test how well the input $\mathbf{x}$ is reconstructed from each latent variable. The notation $\mathcal{L}^{>k}$ highlights that for latent variables $\mathbf{z}^{(>k)}$, the bound is the regular ELBO while for the latent variables $\mathbf{z}^{(\leq k)}$, the bound is evaluated using the (conditional) prior rather than the approximate posterior as the proposal distribution.

4.4.2   A LIKELIHOOD-RATIO SCORE FOR ALL FEATURE LEVELS

While the $\mathcal{L}^{>k}$ bound provides a score for performing semantic OOD detection, it still relies on the data space likelihood function (see equation (4.7) below),

which is known to be problematic for OOD detection (section 4.3.3). To alleviate this, we phrase OOD detection as a likelihood ratio test of being *semantically* in-distribution. A standard likelihood ratio test [74] suggests considering the ratio between the associated likelihoods, which we can approximate on a log-scale by the corresponding lower bounds $\mathcal{L}$ and $\mathcal{L}^{>k}$,

$$\text{LLR}^{>k}(\mathbf{x}) = \mathcal{L}(\mathbf{x}) - \mathcal{L}^{>k}(\mathbf{x}) \,. \tag{4.6}$$

Since, empirically, $\mathcal{L} \geqslant \mathcal{L}^{>k}$, the ratio is always positive as is standard for likelihood ratio tests. A low value of $\text{LLR}^{>k}(\mathbf{x})$ means that the ELBO and $\mathcal{L}^{>k}$ are almost equally tight for the data. On the contrary, a high value indicates that $\mathcal{L}^{>k}$ is looser on the data than the ELBO; hence, the data may be OOD.

We can gather further insights about this score if we write the regular ELBO and the $\mathcal{L}^{>k}$ bounds in the exact form that includes the intractable KL-divergence between the approximate and true posteriors,

$$\mathcal{L} = \log p_\theta(\mathbf{x}) - D_{\text{KL}}\left(q_\phi(\mathbf{z}|\mathbf{x}) \,\|\, p_\theta(\mathbf{z}|\mathbf{x})\right), \tag{4.7}$$
$$\mathcal{L}^{>k} = \log p_\theta(\mathbf{x}) - D_{\text{KL}}\left(p_\theta(\mathbf{z}^{(\leqslant k)}|\mathbf{z}^{(>k)})q_\phi(\mathbf{z}^{(>k)}|\mathbf{x}) \,\|\, p_\theta(\mathbf{z}|\mathbf{x})\right) \,.$$

Subtracting these cancel out the two data likelihood terms $\log p_\theta(\mathbf{x})$ and only the KL-divergences from the approximate to the true posterior remain,

$$\text{LLR}^{>k}(\mathbf{x}) = -D_{\text{KL}}\left(q_\phi(\mathbf{z}|\mathbf{x}) \,\|\, p_\theta(\mathbf{z}|\mathbf{x})\right) \tag{4.8}$$
$$+ D_{\text{KL}}\left(p_\theta(\mathbf{z}^{(\leqslant k)}|\mathbf{z}^{(>k)})q_\phi(\mathbf{z}^{(>k)}|\mathbf{x}) \,\|\, p_\theta(\mathbf{z}|\mathbf{x})\right) \,.$$

Hence, it is clear that compared to the likelihood bound $\mathcal{L}^{>k}$, this likelihood-ratio measures divergence exclusively in the latent space whereas $\mathcal{L}^{>k}$ includes the $\log p_\theta(\mathbf{x})$ term similar to the ELBO. Therefore, the $\text{LLR}^{>k}$ score should be an improved method for semantic OOD detection compared to $\mathcal{L}^{>k}$. Now, it can be noted that if we replace the regular ELBO, $\mathcal{L}$, in (4.7) with the strictly tighter importance weighted bound [69],

$$\mathcal{L}_S^{\text{IWAE}} = \mathbb{E}_{\mathbf{z}_s \sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{1}{N}\sum_{s=1}^{S} \frac{p_\theta(\mathbf{x}, \mathbf{z}_s)}{q_\phi(\mathbf{z}_s|\mathbf{x})}\right], \tag{4.9}$$

then, in the limit $S \to \infty$, we have $\mathcal{L}_S^{\text{IWAE}} \to \log p_\theta(\mathbf{x})$ and the likelihood ratio reduces to

$$\text{LLR}_S^{\geqslant k}(\mathbf{x}) \to D_{\text{KL}}(p_\theta(\mathbf{z}^{(\leqslant k)}|\mathbf{z}^{(>k)})q_\phi(\mathbf{z}^{(>k)}|\mathbf{x}) \,\|\, p_\theta(\mathbf{z}|\mathbf{x})) \tag{4.10}$$

which, in practice, is well-approximated for a finite S. We expect this importance weighted likelihood ratio to monotonically improve upon the one in (4.8) as S increases and the KL-divergence in the regular ELBO that contains terms for which $\mathbf{z}^{(i)}$ is high-dimensional goes to zero.

Since the scores in (4.8) and (4.10) are estimated by sampling their estimators are stochastic objects with nonzero variance. We note that $\mathrm{Var}(\widehat{\mathrm{LLR}}^{>k}) = \mathrm{Var}(\hat{\mathcal{L}}) + \mathrm{Var}(\hat{\mathcal{L}}^{>k}) - 2\,\mathrm{Cov}(\hat{\mathcal{L}}, \hat{\mathcal{L}}^{>k})$. Since $\log p_\theta(\mathbf{x})$ and part of the KL-divergence are identical in the expressions of $\mathcal{L}$ and $\mathcal{L}^{>k}$ we expect $\mathrm{Cov}(\hat{\mathcal{L}}, \hat{\mathcal{L}}^{>k})$ to be positive which reduces the total variance. Empirical results indeed show that $\mathrm{Var}(\widehat{\mathrm{LLR}}^{>k})$ is larger than $\mathrm{Var}(\hat{\mathcal{L}})$ but smaller than $\mathrm{Var}(\hat{\mathcal{L}}^{>k})$. Nevertheless, the variance of the estimators is guaranteed to go to zero as the number of samples is increased.

The OOD scores considered in this research all assume that what discriminates an out-of-distribution from an in-distribution data point are semantic, high-level features. Clearly, if this is not the case and the difference instead lies in low-level statistics, the scores would likely fail. We hypothesize that a complementary bound to (4.5), $\mathcal{L}^{<l}$ described in appendix B.5, might be useful in these cases, but leave further examination to future work.

4.5   EXPERIMENTAL SETUP

Tasks   We follow existing literature [254, 472] and evaluate our method by setting up OOD detection tasks from FashionMNIST [714] to MNIST [380] and from CIFAR10 [356] to SVHN [480]. For each experiment we train our model on the train split of the former dataset and test its ability to recognize the test split of the latter dataset as OOD from the test split of the former dataset. We use the standard train/test splits for the datasets. More details on the datasets can be found in the appendix B.1.

$$q_\phi(\mathbf{z}|\mathbf{x}) \qquad\qquad p_\theta(\mathbf{x}, \mathbf{z})$$
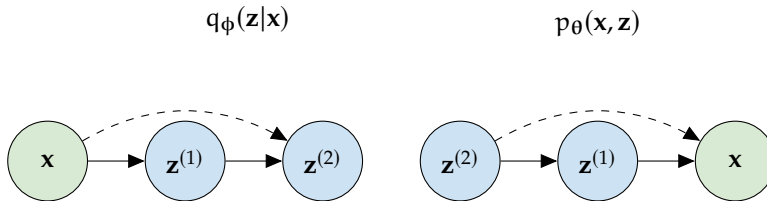
Figure 4.4: The inference and generative models, q_ϕ and p_θ, for an L = 2 layered bottom-up hierarchical VAE as the one used in our experiments. Dashed lines indicate deterministic skip connections which are employed in both networks. Skip connections are found to be useful for optimizing latent variable models [155, 434].

Models   For each OOD task, we train a simple bottom-up hierarchical VAE with L stochastic layers which we will refer to as "HVAE". To alleviate posterior collapse we include skip-connections that connect $\mathbf{z}^{(i)}$ to $\mathbf{z}^{(i+2)}$ for $i \in \{0, L-2\}$ and $\mathbf{z}^{(0)} \equiv \mathbf{x}$ in both the inference and generative models [155] and employ the *free bits* scheme with $\lambda = 2$ [344]. We use weight-normalization [574] on all weights and residual networks in the deterministic paths. A graphical representation of this model can be seen in figure 4.4. We use a Bernoulli output distribution for FashionMNIST/MNIST and a discretized mixture of logistics output distribution [573] for CIFAR10/SVHN. We use $L = 3$ for grayscale images and $L = 4$ for natural images. Full model details are in the appendix B.2.

Baselines   We group baselines into those that use prior knowledge about OOD data, ones that use labels associated with the in-distribution data and purely unsupervised approaches that do not make such assumptions. Our method falls into the latter category. For more information on each baseline, we refer to the original literature.

Evaluation   Following previous work [5, 107, 253, 254, 552] we use the threshold-independent evaluation metrics of Area Under the Receiver Operator Characteristic (AUROC↑), Area Under the Precision Recall Curve (AUPRC↑) and False Positive Rate at 80% true positive rate (FPR80↓) where the arrow indicates the direction of improvement. Note that these metrics are only computable given examples of OOD data but faced with truly OOD data (unknown unknowns), there are many ways to select thresholds to use in practice e.g. as the one that yields a specific tolerable false positive rate on the in-distribution test data. To compute the metrics, we use an equal number of samples from the in-distribution and OOD datasets by including all examples in the smallest of the two sets and randomly sampling equally many from the larger. We compute the $\text{LLR}^{>k}$ score with one and S importance samples denoted by $\text{LLR}_S^{>k}$.

Selection of k To determine whether an example is OOD in practice, the value of $\text{LLR}^{>k}$ is computed on the in-distribution test set for all k and the resulting empirical distribution is used as reference. If for any value of k, the $\text{LLR}^{>k}$ score of a new input differs significantly from the empirical distribution, it is regarded OOD. If it differs for multiple values of k, the value for which it differs the most is selected. In our experiments, we consider an entire dataset at a time and report the results of $\text{LLR}^{>k}$ with the value of k that yielded the highest AUROC↑ for that dataset in a threshold-free manner. In practice, slightly better performance may be achieved by choosing k per example. This would not exclude the use of batching in our method, since $\text{LLR}^{>k}$ is computed after the forward pass.

4.6 RESULTS

The likelihoods for our trained models are in table 4.1 alongside baseline results for in-distribution and OOD data. The main results of the paper on the OOD tasks can be seen along with comparisons to the baseline methods in table 4.2. We note that for all our results, the value of the score ($\mathcal{L}^{>k}$ and LLR$^{>k}$) for the training and test splits of the in-distribution data was observed to have the same empirical distribution to within sampling error hence yielding an AUROC score of ≈ 0.5 as expected. Results on additional commonly used datasets are found in appendix B.7.

Table 4.1: Average bits per dimension of different datasets for models trained on FashionMNIST and CIFAR10. For the hierarchical models we include the $\mathcal{L}^{>k}$ bounds. The likelihoods of training and test splits of the in-distribution data are all cases close. Since we train on dynamically binarized FashionMNIST, our bits/dim are smaller than for Glow. As k is increased for the L$^{>k}$ bound, the bound gets looser, but the model eventually assigns higher likelihood to the in distribution data than to the OOD data. Glow refers to Kingma and Dhariwal [339] and Nalisnick et al. [472]. BIVA refers to our implementation of Maaløe et al. [434].

Method	Dataset	Avg. bits/dim			
		$\log p(x)$	$\mathcal{L}^{>1}$	$\mathcal{L}^{>2}$	$\mathcal{L}^{>3}$
Trained on FashionMNIST					
Glow	FashionMNIST	2.96	-	-	
	MNIST	1.83	-	-	
HVAE (Ours)	FashionMNIST	0.420	0.476	0.579	-
	MNIST	0.317	0.601	0.881	-
Trained on CIFAR10					
Glow	CIFAR10	3.46	-	-	
	SVHN	2.39	-	-	
HVAE (Ours)	CIFAR10	3.74	17.8	54.3	75.7
	SVHN	2.62	10.2	64.0	93.9
BIVA (Ours)	CIFAR10	3.46	8.74	19.7	37.3
	SVHN	2.35	6.62	25.1	59.0

---

[4]Serrà et al. [592] performs the best when high likelihoods are assigned to OOD data such that the overlap with in-distribution data is low. Performance is worse when the overlap is high, cf. Serrà et al. [592, Table 1], as seen with complex images.

Table 4.2: AUROC↑, AUPRC↑ and FPR80↓ for OOD detection for a FashionM-NIST model using scores on the FashionMNIST test set as reference. We bold the best results within the "No OOD-specific assumptions"group since we only compare directly to those. HVAE (ours) refers to our hierarchical bottom-up VAE. BIVA (ours) refers to our implementation of the hierarchical BIVA model.

Method	AUROC↑	AUPRC↑	FPR80↓
FashionMNIST (in) / MNIST (out)			
Use prior knowledge of OOD			
Backgr. contrast. LR (PixelCNN) [552]	0.994	0.993	0.001
Backgr. contrast. LR (VAE) [107]	0.924	-	-
Binary classifier [552]	0.455	0.505	0.886
$p(\hat{y}|\mathbf{x})$ with OOD as noise class [552]	0.877	0.871	0.195
$p(\hat{y}|\mathbf{x})$ with calibration on OOD [552]	0.904	0.895	0.139
Input complexity (S, Glow) [254]	0.998	-	-
Input complexity (S, PixelCNN++) [254]	0.967	-	-
Use in-distribution data labels y			
$p(\hat{y}|\mathbf{x})$ [253, 552]	0.734	0.702	0.506
Entropy of $p(y|\mathbf{x})$ [552]	0.746	0.726	0.448
ODIN [400, 552]	0.752	0.763	0.432
VIB [5, 107]	0.941	-	-
Mahalanobis distance, CNN [552]	0.942	0.928	0.088
Mahalanobis distance, DenseNet [389]	0.986	-	-
Ensemble, 20 classifiers [368, 552]	0.857	0.849	0.240
No OOD-specific assumptions			
- Ensembles			
WAIC, 5 models, VAE [107]	0.766	-	-
WAIC, 5 models, PixelCNN [552]	0.221	0.401	0.911
- Not ensembles			
Likelihood regret [716]	**0.988**	-	-
$\mathcal{L}^{>0}$ + HVAE (ours)	0.268	0.363	0.882
$\mathcal{L}^{>1}$ + HVAE (ours)	0.593	0.591	0.658
$\mathcal{L}^{>2}$ + HVAE (ours)	0.712	0.750	0.548
LLR$^{>1}$ + HVAE (ours)	0.964	0.961	0.036
LLR$_{250}^{>1}$ + HVAE (ours)	0.984	**0.984**	**0.013**
CIFAR10 (in) / SVHN (out)			
Use prior knowledge of OOD			
Backgr. contrast. LR (PixelCNN) [552]	0.930	0.881	0.066
Backgr. contrast. LR (VAE) [716]	0.265	-	-
Outlier exposure [254]	0.984	-	-
Input complexity (S, Glow) [592]	0.950	-	-
Input complexity (S, PixelCNN++) [592]	0.929	-	-
Input complexity (S, HVAE) (Ours) [592][4]	0.833	0.855	0.344
Use in-distribution data labels y			
Mahalanobis distance [389]	0.991	-	-
No OOD-specific assumptions			
- Ensembles			
WAIC, 5 models, Glow [107]	1.000	-	-
WAIC, 5 models, PixelCNN [552]	0.628	0.616	0.657
- Not ensembles			
Likelihood regret [716]	0.875	-	-
LLR$^{>2}$ + HVAE (ours)	0.811	0.837	0.394
LLR$^{>2}$ + BIVA (ours)	**0.891**	**0.875**	**0.172**

4.6.1   LIKELIHOOD-BASED OOD DETECTION

(a)

(b)

(c)
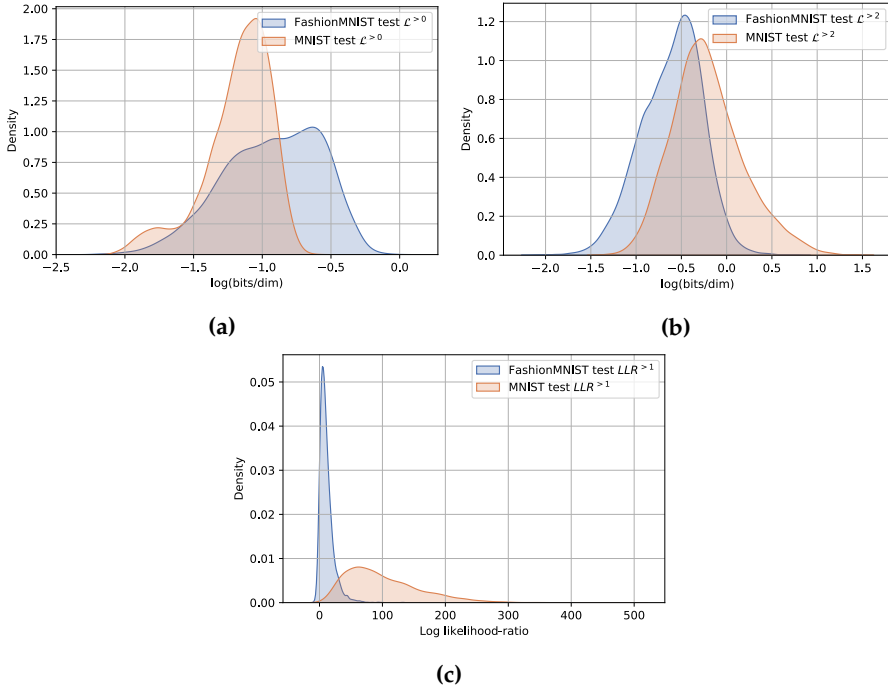
Figure 4.5: Empirical densities of FashionMNIST (in-distribution) and MNIST (OOD) using the raw likelihood (a), the $\mathcal{L}^{>2}$ bound (b) and the LLR$^{>1}$ score (c). All densities are computed using the HVAE model. For the regular likelihood MNIST is very clearly more likely on average than the FashionMNIST test data while with the $\mathcal{L}^{>2}$ bound separation is better but significant overlap remains. The LLR$^{>1}$ provides a high degree of separation. Likelihoods are reported in units of the natural log of the number of bits per dimension.

We first report the results of the different variations of the $\mathcal{L}^{>k}$ bound for OOD detection. We reconfirm the results of Nalisnick et al. [472] by observing that our hierarchical latent variable models also assign higher $\mathcal{L}^{>0}$ to the OOD dataset in the FashionMNIST/MNIST and CIFAR10/SVHN cases resulting in an AUROC↑ inferior to random (table 4.2). Switching the in-distribution data for the OOD data in both cases result in correctly detecting the OOD data; an asymmetry also reported by Nalisnick et al. [472]. Figure 4.5(a) shows the density of $\mathcal{L}^{>0}$ in bits per dimension [636] by the model trained on FashionMNIST when evaluated on the FashionMNIST and MNIST test sets. We observe a high degree

of overlap, with less separation of the OOD data compared to similar results of autoregressive and flow-based models, like Xiao, Yan, and Amit [716].

We then evaluate the looser $\mathcal{L}^{>k}$ (4.5) for $k \in \{1, L\}$. Figure 4.5(b) shows the result for $\mathcal{L}^{>2}$, which yielded the highest AUCROC↑, only slightly better than random. Like Maaløe et al. [434], we see that increasing the value of k generally leads to improved OOD detection. However, we also observe that the two empirical distributions never cease to overlap. Importantly, depending on the OOD dataset, the amount of remaining overlap can be high which limits the discriminatory power of the likelihood-based $\mathcal{L}^{>k}$ bound. This is in-line with the pathological behavior of the raw likelihood of latent variable models when used for OOD detection [716]. Since a high degree of overlap also seems present in Maaløe et al. [434], and we see the same problem for our BIVA model trained on CIFAR10, we do not expect this to be due to the less expressive HVAE.

4.6.2 LIKELIHOOD-RATIO-BASED OOD DETECTION

We now move to the likelihood ratio-based score. We find that LLR$^{>k}$ separates the OOD MNIST data from in-distribution FashionMNIST to a higher degree than the likelihood estimates as can be seen by the empirical densities of the score in figure 4.5(c). We note that the likelihood ratio between the ELBO and the $\mathcal{L}^{>k}$ bound provides the highest degree of separation of MNIST and FashionMNIST as measured by the AUROC↑ for $k = 1$ smaller than L. This is not surprising since the value of k that provides the maximal separation to the reference in-distribution dataset need not be the one for which $\mathcal{LLR}^{>k}$ is overall maximal for the OOD dataset. We also visualize the ROC curves resulting from using the LLR$^{>k}$ score for OOD detection on both FashionMNIST/MNIST and CIFAR10/SVHN and compare it to the ROC curves resulting from the different $\mathcal{L}^{>k}$ bounds in figure 4.6, respectively. On both datasets we see significantly better discriminatory performance when using the LLR$^{>k}$ score.

Table 4.2 shows that BIVA improves upon the HVAE model for OOD detection on CIFAR while table 4.1 shows that the BIVA model also improves upon the HVAE in terms of likelihood. We hypothesize that models larger than our implementation of BIVA, with better likelihood scores may perform even better [102, 434, 657].

4.6.3 COMPARISON TO BASELINES

Performance   Table 4.2 summarize our results compared to baselines based on the commonly used AUROC↑, AUPRC↑ and FPR80↓ metrics. Our method outperforms other generative model-based methods such as WAIC [107] with Glow model and performs similarly to the likelihood regret method of [716]. Further-
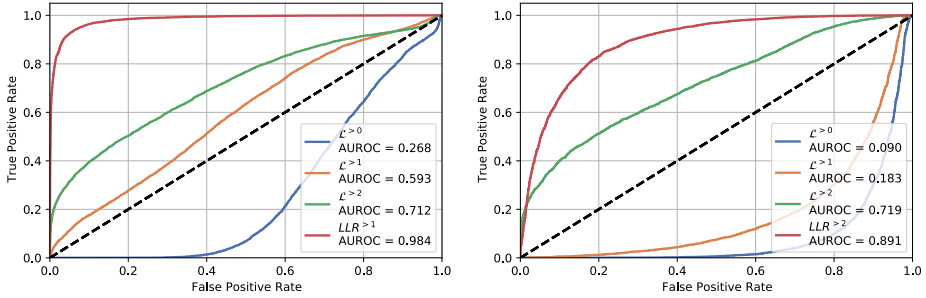
Figure 4.6: ROC curves with AUROC score for detecting MNIST as OOD with the HVAE model trained on FashionMNIST (left) and SVHN as OOD with the BIVA model trained on CIFAR10 (right). A ROC curve is plotted for each of the $\mathcal{L}^{>k}$ bounds including the ELBO along with one for the best-performing log likelihood-ratio LLR$^{>1}$.

more, our method performs similarly to the background contrastive likelihood ratio method of Ren et al. [552] on FashionMNIST/MNIST but contrary to the failure of that method on CIFAR10/SVHN reported by [716], our method performs very well on this task too. Our approach outperforms all supervised approaches that use in-distribution labels or synthetic examples of OOD data derived from the in-distribution data including ODIN [400] and the predictive distribution of a classifier $p(\hat{y}|\mathbf{x})$ trained and evaluated in various ways (see Ren et al. [552]).

Runtime   For a full evaluation of a single example across all feature levels of a model with L stochastic layers, our method requires $L-1$ forward passes through the inference and generative networks as well as computing the likelihood ratio, of which the forward passes are dominant. For a typical forward pass that is linear in the input dimensionality, D, and the number of stochastic layers, L, this amounts to computation of $\mathcal{O}(DL)$. Compared to some related work that either requires an $M > 1$ sized batch of inputs of which either all or none are OOD [473] or cannot be applied to batches due to the required per-example optimization [716], our method additionally is applicable to batches of any size that may consist of both OOD and in-distribution examples which provides drastic speedups via vectorization and parallelization. Furthermore, the method of Xiao, Rasul, and Vollgraf [714] requires refitting the inference network of a VAE which can be computationally demanding. Compared to the likelihood ratio proposed in Ren et al. [552], our method requires training only a single model on a single dataset.

4.7 Discussion

Deep generative models are state-of-the-art density estimators, but the OOD failures reported in recent years have raised concerns about the limitations of such density estimates. Recent work on improving OOD detection has largely sidestepped this concern by relying on additional assumptions that strictly should not be needed for models with explicit likelihoods. While the engineering challenge of building reliable OOD detection schemes is important, it is of more fundamental importance to understand *why* the naive likelihood test fails. We have provided evidence that low-level features of the neural nets dominate the likelihood, which gives a *cause* to the *why*. The fact that a simple score for measuring the importance of semantic features yield state-of-the-art results on OOD detection without access to additional information gives validity to our hypothesis.

The findings from, amongst others, Nalisnick et al. [472] and Serrà et al. [592] have a clear relation to information theory and compression. Semantically complex in-distribution data yields models with diverse low-level feature sets that enable generalization across datasets. Simpler datasets can only yield models with less diverse low-level feature sets compared to complex training data. Hence, there can be an asymmetry where the likelihoods of simple OOD data can be high for a model trained on complex data, but not the other way around. Loosely put, the minimal number of bits required to losslessly compress data sampled from some distribution is the entropy of the generating process [439, 596]. Townsend, Bird, and Barber [648] recently showed that VAEs can be used for lossless compression at rates superior to more generic algorithms.

We also note that since the hierarchical VAE is a probabilistic graphical latent variable model, it lends itself very naturally to manipulation at the feature level [343, 435, 436]. This property sets it apart from other generative models that do not explicitly define such a hierarchy of features. This in turn enables reliable OOD detection with our methodology while making no explicit assumptions about the nature of OOD data and only using a single model. This has not been achieved with autoregressive or flow-based models.

4.8 Conclusion

In this paper we study unsupervised out-of-distribution detection using hierarchical variational autoencoders. We provide evidence that highly generalizable low-level features contribute greatly to estimated likelihoods resulting in poor OOD detection performance. We proceed to develop a likelihood-ratio based score for OOD detection and define it to explicitly ensure that data must be in-distribution across all feature levels to be regarded in-distribution. This ratio is mathematically shown to perform OOD detection in the latent space of the

model, removing the reliance on the troublesome input-space likelihood. We point out that contrary to much recent literature on OOD detection, our approach is fully unsupervised and does not make assumptions about the nature of OOD data. Finally, we demonstrate state-of-the-art performance on a wide range of OOD failure cases.

ACKNOWLEDGEMENTS

MODEL-AGNOSTIC OUT-OF-DISTRIBUTION DETECTION USING COMBINED STATISTICAL TESTS

This chapter is a piece of original research by the candidate, previously published elsewhere and reprinted here with permission:

[B] Bergamin, F., Mattei, P.-A., **Havtorn, J. D.**, Senetaire, H., Schmutz, H., Maaløe, L., Hauberg, S., Frellsen, J., "Model-Agnostic Out-of-Distribution Detection Using Combined Statistical Tests". In: *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*. volume 151. Valencia, Span: PMLR, 2022. arXiv: 2203.01097 [coauthor] [45]

ABSTRACT

We present simple methods for out-of-distribution detection using a trained generative model. These techniques, based on classical statistical tests, are model-agnostic in the sense that they can be applied to any differentiable generative model. The idea is to combine a classical parametric test (Rao's score test) with the recently introduced typicality test. These two test statistics are both theoretically well-founded and exploit different sources of information based on the likelihood for the typicality test and its gradient for the score test. We show that combining them using Fisher's method overall leads to a more accurate out-of-distribution test. We also discuss the benefits of casting out-of-distribution detection as a statistical testing problem, noting in particular that false positive rate control can be valuable for practical out-of-distribution detection. Despite their simplicity and generality, these methods can be competitive with model-specific out-of-distribution detection algorithms without any assumptions on the out-distribution.

5.1 INTRODUCTION

The ability to recognize when data are anomalous, i.e. if they originate from a distribution different from that of the training data, is a necessary property for machine learning models for safe and reliable applications in the real world. Historically, Bishop [50] proposed to use a one-sided threshold on the log-likelihoods of a learned model as a decision rule to identify outliers in a dataset. However,

recently, Hendrycks, Mazeika, and Dietterich [254] and Nalisnick et al. [472] showed that state-of-the-art deep generative models (DGMs) failed in this task, assigning higher a likelihood to out-of-distribution (OOD) data than indistribution data. Most of the recent works focused on proposing new test statistics to alleviate the problem of using the plain likelihood, see section 5.5 for details.

We believe that OOD detection should be formulated as statistical hypothesis testing [2, 237, 473]. Since the power of a single test depends on the out-distribution [737], we propose to approach this problem by using a combination of multiple statistical tests. While the power of the combined test also depends on the out-distribution, we hypothesize that the combined test empirically will perform better, especially in situations where one of the statistics fails. Furthermore, the use of the statistical testing framework has several advantages. Since we obtain a p-value, it is more natural deciding on a threshold as this corresponds to the significance level. In addition to that, it also allows us to correct for the multiple comparisons problem when identifying outliers in a dataset by controlling the number of Type I errors through the false discovery rate (FDR).

In summary, our contributions are the following:

- We illustrate the benefits of combining multiple statistical tests to perform OOD detection with DGMs using well-established methods. This allows for a proper decision procedure to control the FDR in a real outlier detection setting.

- We revisit some proposed detection scores and highlight their alternative formulation as classical significance tests.

- Empirically we show the complementarity of the typicality and the score statistics and that their combination leads to a robust score for anomaly detection.

5.2   USING STATISTICAL TESTS FOR OUT-OF-DISTRIBUTION DETECTION

We consider some data of interest that live in a space $\mathcal{X}$. Assume that we have a curated dataset $x_1, \ldots, x_m$, i.e. there are no outliers, and we are interested in understanding if some new data $\tilde{x}_1, \ldots, \tilde{x}_n$ are collectively anomalies. In other words, we wonder whether or not $\tilde{x}_1, \ldots, \tilde{x}_n$ are likely to come from the same distribution that generated our curated dataset. We present in this section two different approaches for doing out-of-distribution detection using statistical tests: one based on classical parametric tests and one based on maximum mean discrepancy. A convenient property of the tests we consider is that they are all one-sided, which means we can expect them to be larger when the data are more

likely to be OOD. This allows us to compute p-values by simply using the empirical CDF, which is hyperparameter-free.

Note that in this problem formulation, the case $n = 1$ corresponds to the situation where we need to decide if a *single* data point is out-of-distribution. This hardest setting will be of particular interest, and this is also the main focus of recent work, see section 5.5.

5.2.1 Parametric tests for out-of-distribution detection

The typical approach is to consider a parametric family $(p_\theta)_{\theta \in \Theta}$ of probability densities over $\mathcal{X}$ and learn a suitable $\theta_0 \in \Theta$ using any inference technique, for example maximum likelihood, and the clean data $x_1, \ldots, x_m$. Depending on the input domain, $(p_\theta)_{\theta \in \Theta}$ could be composed of DGMs (in that case, θ would be neural network weights) or Gaussian mixture models (in that case, θ would be composed of means, covariances, and proportions). The question we wish to answer may then be phrased: *is p_{θ_0} an appropriate model for $\tilde{x}_1, \ldots, \tilde{x}_n$?*

We choose to formalize this problem as a *parametric test* whose alternative hypothesis is that $\tilde{x}$ is *out-of-distribution*. More specifically, if we assume that $\tilde{x}_1, \ldots, \tilde{x}_n \sim_{\text{i.i.d.}} p_{\tilde{\theta}}$ for some unknown $\tilde{\theta} \in \Theta$, we wish to test $\mathcal{H}_0 : \tilde{\theta} = \theta_0$ against $\mathcal{H} : \tilde{\theta} \neq \theta_0$, where the alternative hypothesis $\mathcal{H}$ is that the test points are OOD.

Many tests have been proposed for this purpose. The three most famous are the *likelihood ratio test* of Neyman and Pearson [482], Rao's (1948) *score test*, and *the Wald test* [673]. These three classics are nicely reviewed by Buse [74] or by Rao [546], who called them the "Holy Trinity". A recent and interesting one is the *gradient test* of Terrell [635], which is reviewed in great detail in Lemonte's (2016) monograph.

Let us review the statistics of these four tests:

- likelihood ratio statistic is $S_{LR} = 2(\ell(\hat{\theta}) - \ell(\theta_0))$,

- Wald statistic is $S_W = (\hat{\theta} - \theta_0)^\top I(\hat{\theta})(\hat{\theta} - \theta_0)$,

- score statistic is $S_S = \nabla\ell(\theta_0)^\top I(\theta_0)^{-1}\nabla\ell(\theta_0)$,

- gradient statistic is $S_G = \nabla\ell(\theta_0)^\top(\hat{\theta} - \theta_0)$,

where $\ell(\theta) = \log p_\theta(\tilde{x}_1, \ldots, \tilde{x}_n)$ is the likelihood function, $I(\theta) = \mathbb{E}_{p_\theta}[\nabla\ell(\theta)\nabla\ell(\theta)^\top]$ is the Fisher information matrix (FIM), and $\hat{\theta} \in \arg\max_{\theta \in \Theta} \ell(\theta)$.

The likelihood ratio statistic, the Wald statistic and the gradient statistic all require to fit a model on the additional data points $\tilde{x}_1, \ldots, \tilde{x}_n$ in order to compute either $\ell(\hat{\theta})$ or $\hat{\theta}$. In our setting, if we want to use one of those statistics as an OOD score for a single example, we should fit a DGM on that single data point. Xiao, Yan, and Amit [716] did this for a variational autoencoder (VAE, [340, 557]) by

only re-fitting inference network (or encoder) to the additional example, which is a typical approach to dealing with out-of-sample data in VAEs, as argued by Cremer, Li, and Duvenaud [136] and Mattei and Frellsen [448]. However, much of the recent works in the literature [552, 581, 592] mainly focus on deriving different versions of what they call a likelihood ratio statistic.

We tried to derive a general way to compute both the Wald statistic and the gradient statistic, by computing $\hat{\theta}$ with a few steps of a gradient-based optimization algorithm initialized at θ_0, but this resulted in a very unstable update leading to computational issues (results not shown). Therefore, in this work we focus on studying the relevance of the score statistic for performing out-of-distribution detection since it is the only statistic that does not require fitting an additional model to the OOD data.

5.2.2   MAXIMUM MEAN DISCREPANCY FOR OUT-OF-DISTRIBUTION DETECTION

Another way of approaching out-of-distribution detection from a testing perspective is through a *two-sample test*. Denoting p_{data} the true training data distribution, the goal is to test $\mathcal{H}_0 : \tilde{x}_1, \ldots, \tilde{x}_n \sim p_{data}$ against $\mathcal{H} : \tilde{x}_1, \ldots, \tilde{x}_n \nsim p_{data}$, where the alternative hypothesis $\mathcal{H}$ again is that the test points are OOD.

A popular way of building statistics for two-sample tests is to use a measure of distance between p_{data} and the distribution of $\tilde{x}_1, \ldots, \tilde{x}_n$. The key idea here will be to use the trained generative model to build this measure of distance. To this end, we will use the *maximum mean discrepancy (MMD)* of Gretton et al. [219], which is a kernel-based measure of distance. Then, p_θ will be used to specify an appropriate kernel.

More specifically, given a kernel whose feature map is $\Phi : \mathcal{X} \to \mathcal{H}$, the MMD between two distributions P and Q over $\mathcal{X}$ is defined as

$$\text{MMD}_\Phi(P, Q) = \|E_{X \sim P}[\Phi(X)] - E_{Y \sim Q}[\Phi(Y)]\|_{\mathcal{H}} \ . \tag{5.1}$$

In our context, the test statistics will be of the form

$$\text{MMD}_\Phi \left(\frac{1}{m} \sum_{i=1}^{m} x_i, \frac{1}{n} \sum_{i=1}^{n} \tilde{x}_i \right) = \left\| \frac{1}{m} \sum_{i=1}^{m} \Phi(x_i) - \frac{1}{n} \sum_{i=1}^{n} \Phi(\tilde{x}_i) \right\|_{\mathcal{H}} , \tag{5.2}$$

where Φ is a kernel feature map built using the generative model and $x_1, \ldots, x_m$ is the training data, i.e. samples from p_{data}. When $\mathcal{H}$ is a simple finite-dimensional Hilbert space and Φ can be computed easily, then (5.2) can be computed by going through the data and computing the means in an online fashion.

As always with kernel methods, a key question is how to choose the kernel, or its feature map Φ. Here, we want to use the trained generative model p_θ to build our kernel feature map Φ.

The Fisher kernel   An important example of kernel based on a generative model is the *Fisher kernel* of Jaakkola and Haussler [294]. The embedding of this kernel is the Fisher score

$$\Phi_{\text{Fisher}}(x) = I(\theta)^{-\frac{1}{2}} \nabla \log p_\theta(x) \ , \tag{5.3}$$

and the corresponding reproducing kernel Hilbert space norm is just the ℓ_2 norm: $|| \cdot ||_{\mathcal{H}} = || \cdot ||_2$. In the case of the Fisher kernel, this means that (5.2) becomes:

$$\text{MMD}_{\Phi_{\text{Fisher}}} \left(\frac{1}{m} \sum_{i=1}^{m} x_i, \frac{1}{n} \sum_{i=1}^{n} \tilde{x}_i \right) =$$

$$\left\| \frac{I(\theta)^{-\frac{1}{2}}}{m} \sum_{i=1}^{m} \nabla \log p_\theta(x_i) - \frac{I(\theta)^{-\frac{1}{2}}}{n} \sum_{i=1}^{n} \nabla \log p_\theta(\tilde{x}_i) \right\|_2 \ . \tag{5.4}$$

We will see later that MMD with a Fisher kernel is closely related to the score statistic. In appendix D.2, we additionally show that another popular OOD metric known as the *Mahalanobis score* [389] can be interpreted as a MMD statistic with a certain Fisher kernel.

The typicality kernel   A very simple approach of embedding the data using p_θ is to choose $\Phi_{\text{Typical}}(x) = \log p_\theta(x)$. Then, MMD is exactly equivalent to the *typicality test statistic* of Nalisnick et al. [473], although this connection was not explicitly stated by Nalisnick et al. [473]. Because of this, we call the kernel $k(x, y) = \log p_\theta(x) \cdot \log p_\theta(y)$ the *typicality kernel*. While Φ_{Typical} is not as well motivated as a kernel as Φ_{Fisher}, the concepts of typicality and typical set can be used to explain unintuitive behaviors of probability distributions in high-dimensional space as highlighted by Nalisnick et al. [472]. We also found that using this kernel generally gives good results for OOD tasks. An interesting analysis that we'd not consider in this paper would be to study the properties of this kernel.

In general, neither of these two kernels are characteristic, meaning that our MMD can be zero even if the distributions are not identical. This could be solved by combining them with a characteristic kernel, as in Liu et al. [412], at the price of including a new hyperparameter.

5.3   Combining different test statistics

For single-sample OOD detection, Zhang, Goldstein, and Ranganath [737] proved that there is not a single statistic that is constantly better compared to all the possible alternatives of interest. For this reason, we believe that using a combination of different test statistics should lead to an overall better OOD detection in settings

where a single statistic might fail. Assume we compute k different test statistics $T_1, \ldots, T_k$, each testing $\mathcal{H}_0$ against $\mathcal{H}$ as defined in sections 5.2.1 and 5.2.2. The goal is to combine these different tests into a single statistical test that ideally will perform better than the initial single tests. However, different tests can have different magnitudes, and they can differ also in the direction of out-of-distribution detection, i.e. for some statistics having a higher value is associated with being OOD, while for other smaller values are OOD. This makes a combination non-trivial.

Morningstar et al. [467] proposed the density of states estimator (DoSE) to overcome this problem. They only focused on the single-sample detection task, i.e. $n = 1$ following our problem formulation. Their idea is to fit different non-parametric density estimators, such as a kernel-density estimator (KDE) or a one-class support vector machines (SVM), for each different statistic $T_1, \ldots, T_k$ by using the values computed on the training set examples. For a single test example, $\tilde{x}_1$, they first compute $T_1, \ldots, T_k$ and then combine those statistics by summing the different KDEs log-density. While this approach can be used for any type of statistic, and thus is more general, it uses less prior information. Indeed, if we use only statistics that are truly one-sided, then we assume that a method that leverages the true nature of the statistics should work better. In addition to that, fitting a KDE introduces an additional hyperparameter.

In our work, instead, we propose a different approach and leverage the fact that we use only one-sided test statistics. This setting is a well-studied problem in the literature both for independent [184, 186] and dependent one-sided test statistics [65, 703]. All these approaches rely on the computation of p-values of each statistic for the test set $\tilde{x}_1, \ldots, \tilde{x}_n$. This corresponds to computing $p_j = \Pr(T_j > t_j \mid \mathcal{H}_0)$, i.e. the probability that the j'th test is bigger than the observed value under the null hypothesis $\mathcal{H}_0$, where we assume that each T_j has a continuous distribution. Using p-values also solves the problem of the statistics having different scales. Indeed, p-values transform the different test statistics into the unit interval.

Computation of p-values   We want to approximate the distribution of the p-values $p_1, \ldots, p_k$ of $\tilde{x}_1, \ldots, \tilde{x}_n$ under the null hypothesis $\mathcal{H}_0$. When $\mathcal{H}_0$ is true, then p_j is uniformly distributed on the interval $[0, 1]$. To succeed in this, we should be able to compute $p_j = \Pr(T_j > t_j \mid \mathcal{H}_0)$, therefore we need to estimate the distribution of each statistic T_j under $\mathcal{H}_0$. As done by Nalisnick et al. [473], we assume the existence of a validation set $\mathbf{X}'$ that was not used to train our generative model. From $\mathbf{X}'$ we bootstrap S new datasets $\{\mathbf{X}'_s\}_{s=1}^S$ of size M' by using bootstrap resampling. When n is small, for example $n = 1$ or $n = 2$, where $n = 1$ corresponds to single-sample OOD detection, and the validation set is big, a convenient alternative to bootstrapping is to directly evaluate each test statistic T_j on

every single validation example. Asymptotically, this is equivalent to creating S new datasets of size $M' = 1$ when $S \to \infty$. In case of $n = 2$, i.e. two-samples OOD detection, and a big validation set we can simply bootstrap without resampling. We then use these values to estimate the empirical distribution function (eCDF) of the considered statistic T_j under $\mathcal{H}_0$. To obtain the p-values of test examples $\tilde{x}_1, \ldots, \tilde{x}_n$ for the test statistic $T_j = t_j$, we simply compute $p_j = 1 - \Pr(T_j < t_j \mid \mathcal{H}_0)$ using the eCDF.

Combining test statistics by combining p-values   Fisher's (1925) method is a procedure to combine different p-values $p_1, \ldots, p_k$. This method assumes that all the considered test statistics are independent, and Folks and Little [186] proved that it is asymptotically optimal among all methods of combining independent tests. Given $T_1, \ldots, T_k$ and corresponding p-values $p_1, \ldots, p_k$, Fisher's method combines the p-values into a test statistic X^2 defined as

$$X^2 \sim -2 \sum_{j=1}^{k} \ln(p_j) \ . \tag{5.5}$$

In case all null-hypotheses are accepted, the resulting test statistic X^2 follows a chi-squared distribution with $2k$ degrees of freedom. In the appendix D.4.2, we also consider the Harmonic mean p-value [703] as a way to combine p-values from different statistics. This method usually works best when the statistics are not independent.

5.4   From test statistics to practical out-of-distribution scores

Several of the test statistics that we consider make use of the inverse of the Fisher information matrix $I(\theta)$. The true Fisher information matrix requires an identifiable model to be invertible [694] and computing its inverse is $O(m^3)$, where m is the number of model parameters. For DGMs, the Fisher information matrix might not be invertible due to the fact that DGMs typically do not satisfy the identifiability condition. Also, the inversion may be computationally impractical, since state-of-the-art DGMs involve very high-dimensional parameter spaces Θ. For the same reason, storing $I(\theta)$ can also be challenging.

We replace it by using a proxy matrix that has to be easy to compute and invert. A first idea is to simply replace $I(\theta)$ by the identity matrix. A more refined way is to look for a diagonal approximation. In appendix D.1, we describe cheap ways of computing such approximations. In particular, we will study two cases: the case where $I(\theta)$ is replaced by the identity matrix and the case where $I(\theta)$ is replaced by a diagonal matrix estimated using the training data.

A possible third option would be to estimate the diagonal of $I(\theta)$ using samples from the model. However, for autoregressive models as the PixelCNN, sampling is a sequential procedure, and therefore it is computationally expensive to generate many samples when the input-space is high-dimensional. For this reason, we do not consider it in this work. More complex and precise approximations of the FIM exists, such as the Kronecker-factored Approximate Curvature (K-FAC, [444]), but these are not defined for all types of layers used by state-of-the-art models.

On the difficulty of computing per-example gradients   Both the diagonal approximation of the FIM and the computation of the MMD with Fisher kernel of (5.4) require the gradient computation for all training and test examples. This is known as a costly procedure. For example, if we have to compute the gradient for N examples using a simple fully connected network with l layers of size p, the naive procedure of using a batch-size of dimension 1 is $O(Nlp^2)$ [210]. While more efficient per-example gradient computations were proposed [210, 560], these techniques can only be applied on simple fully connected or convolutional networks. While for this paper we relied on the naive solution of looping through every example one at the time, a more efficient solution is provided by the BackPACK library [142] which allows to compute the gradient with respect each sample in a mini-batch.

5.4.1 RELATIONSHIP BETWEEN MMD WITH FISHER KERNEL AND THE SCORE STATISTIC AND GRADIENT NORM

Depending on the choice of the Fisher information approximation, we can notice that there is a strong connection between the MMD using a Fisher kernel, the score statistic and the gradient norm in terms of expected OOD performance. Let us start by looking at the case where we approximate $I(\theta)$ with a diagonal matrix estimated using the training data. At the maximum likelihood estimate, we have that $\mathbb{E}[\nabla \log p_\theta(x)] = 0$, i.e. the first term inside the norm is 0. Therefore, we expect that the differences between the OOD scores computed by using (5.4) will be preserved if we only consider $\left\| I(\theta)^{-1/2} \nabla \log p_\theta(\tilde{x}_1, \ldots, \tilde{x}_n) \right\|_2$, which corresponds to the square root of the score statistic. Since taking the square root still preserves the difference between values, we can expect that the MMD using a Fisher kernel will perform closely to the score statistic. The same reasoning also holds in case we replace the FIM with an identity matrix. In this specific case, instead, we will get that $\left\| I(\theta)^{-1/2} \nabla \log p_\theta(\tilde{x}_1, \ldots, \tilde{x}_n) \right\|_2 = \left\| \nabla \log p_\theta(\tilde{x}_1, \ldots, \tilde{x}_n) \right\|_2$, which corresponds to considering the gradient norm.

Computationally speaking, considering the score statistic instead of the MMD Fisher lets us avoid going through the entire training set to compute the average

gradient (first term in (5.4)) while carrying the same information. Therefore, in this paper, we will mainly focus on the combination of the typicality test and the score statistic.

5.4.2 Why does it make sense to combine the score statistic and the typicality test?

Let us discuss our choice of combining the score statistic and the typicality test. We will try to look in which situations one of the test fails and the other works and vice versa. Both examples assume that the in-distribution data follows an $\mathcal{N}(0, I_D)$ distribution, and that the correct model has been learned by fitting $(\mathcal{N}(\theta, I_D))_{\theta \in \mathbb{R}^D}$ via maximum-likelihood. Even in this simple setting with no model misspecification, we will see that the two statistics that we consider may have very different strengths.

In this simple Gaussian case, the score statistic can be computed exactly and will be $||\tilde{x}_1 + \ldots + \tilde{x}_n||_2^2$. On the other hand, the typicality statistic will be $|(||\tilde{x}_1||_2^2 + \ldots + ||\tilde{x}_n||_2^2)/(2 \cdot n) - D/2|$. One interesting regime is the very high-dimensional one ($D \to \infty$). Indeed, by the law of large numbers, these random statistics become deterministic quantities.

Typicality fails, the score succeeds   Assume that we have two independent OOD data samples that follow a product of truncated normal distributions, with density proportional to

$$\mathcal{N}(x|0, I_D) \cdot \mathbf{1}\{x_1 > 0, \ldots, x_D > 0\} \ . \tag{5.6}$$

We denote by $T_{\text{score}}^{\text{ood}}$, $T_{\text{score}}^{\text{id}}$ and $T_{\text{typicality}}^{\text{ood}}$, $T_{\text{typicality}}^{\text{id}}$ the statistics obtained when confronted with either OOD data from the truncated normal, or the in-distribution data. While these statistics are random in general, they will become deterministic when $D \to \infty$, by virtue of the law of large numbers.

For the typicality statistic, these two OOD samples will be indistinguishable from Gaussian ones. Indeed, when $D \to \infty$, both $T_{\text{typicality}}^{\text{ood}}$ and $T_{\text{typicality}}^{\text{id}}$ will be $\mathcal{O}(D)$. On the other hand, for the score, one can show that

$$T_{\text{score}}^{\text{ood}} - T_{\text{score}}^{\text{id}} \sim 2D\mu_{\text{TN}}^2 \ , \tag{5.7}$$

where $\mu_{\text{TN}} > 0$ is the mean of the truncated normal distribution.

Typicality succeeds, the score fails   Let us now consider as the OOD distribution a Dirac distribution with mean 0. Suppose that we see a single sample from this distribution. In this case, the score statistic will be 0, and will therefore not

detect that the point is actually OOD. However, when D is large, the typicality test will be able to declare that this point is anomalous, as shown by Nalisnick et al. [473].

Therefore, we have that the typicality test and the score statistic are complementary and measure a different type of information. In appendix D.4.1, we empirically show that they are not correlated, by plotting the two measures against each other and by computing the correlation matrix.

5.5   RELATED WORKS

Since Nalisnick et al. [472] and Hendrycks, Mazeika, and Dietterich [254], different test statistics or methodologies for OOD detection using DGMs were proposed. Most of the recent solutions were highly influenced by three major lines of work: *typicality set*, *likelihood ratio* test statistics, and *model misestimation*.

The typicality set hypothesis was introduced by Nalisnick et al. [473] as a possible explanation for the DGMs assigning higher likelihood to OOD data. The typicality set is the subset of the model full support where the model samples from and this does not intersect with the region of higher likelihood. While the typicality test was introduced for batch OOD detection, Morningstar et al. [467] shows that it also works well in the single-sample case. This is also confirmed by our own experiments.

The likelihood ratio test statistic method by Ren et al. [552] assumes that every input is composed by a background component and a semantic component. For OOD detection, only the semantic component matters. In addition to a model trained on the in-distribution data, they proposed to train a background model on perturbed inputs data and then for each test example consider as OOD score the likelihood ratio between the two models. Schirrmeister et al. [581], instead, trained the background model on a more general distribution of images by considering 80 million general tiny images. Similarly to these approaches, Serrà et al. [592] argued that the failure of DGMs is due to the high-influence that the input complexity has on the likelihood. Therefore, they proposed to use a general lossless image compression algorithm as a background model. All these methods, however, require additional knowledge of the OOD data for either choosing an image augmentation procedure to perturb the input data or for choosing a specific compressor.

Another line of works blame the models themselves and not the test statistics. Zhang, Goldstein, and Ranganath [737] argued that model misestimation is the main cause of higher likelihood assigned to OOD data. This can be due to both the model architecture and the maximum likelihood objective. Kirichenko, Izmailov, and Wilson [347] and Schirrmeister et al. [581] showed that normalizing flows can achieve better OOD performance despite achieving a worse likelihood

if one changes some model design choices. Other works in the literature focused on deriving specific test statistics that work only for a specific model, for example for VAEs [245, 434, 716], or for normalizing flows [2, 347].

As mentioned in the introduction, we frame the OOD detection problem in terms of a statistical test problem. Recently, Haroush et al. [237] showed that adopting hypothesis testing at the layer and channel level of a neural network can be used for OOD detection in the discriminative setting. They used both Fisher's method and Simes' method to combine class-conditional p-values computed for each convolutional and dense layer of a deep neural network. We focus on the unsupervised setting using DGMs and use hypothesis testing on statistics that can be computed on all differentiable DGM. As already explained in section section 5.3, Morningstar et al. [467] considered the combination of different statistics for OOD detection. The main difference with their approach is that we propose statistics that can be applied to any differentiable generative model and combine them by using Fisher's method, which takes advantage of using only one-sided independent statistics. Concurrently, Choi et al. [108] derived the score statistic by starting from the likelihood ratio statistic and applying a Laplace approximation. They computed the score statistic only for certain layers of the model and for a specific example, the OOD score is given by the infinity norm of these different layer scores after a ReLU operation. Our procedure differs both in the derivation of the score statistic and its usage since we compute the score statistic for the entire model.

5.6 Experimental Setup

To evaluate the performance of the combination of the typicality test and the score statistic in detecting OOD data, we follow the experiments of Hendrycks, Mazeika, and Dietterich [254] and Nalisnick et al. [472] and considered the OOD detection task on three image dataset pairs that have been proven challenging for DGMs, i.e. FashionMNIST [714] vs MNIST [375], CIFAR10 [356] vs SVHN [480], and CIFAR10 vs CIFAR100. Winkens et al. [704] divide these tasks into *far*-OOD tasks, where the in-distribution and out-distribution are different such as in the case of CIFAR10 against SVHN, and *near*-OOD where the two distributions are pretty similar, such as CIFAR10 and CIFAR100. *Near*-OOD tasks are usually most challenging.

For each task, we trained three different state-of-the-art DGMs, a PixelCNN++ [573], a Glow model [339], and a hierarchical variational autoencoder [340, 557] with bottom-up inference (HVAE, [69]). These are DGMs parametrized by neural networks that make different assumptions in the modelling choice of the target distribution. In addition to that, for PixelCNN++ and Glow we have a tractable likelihood while for HVAE we can only estimate a lower bound. A more in-depth
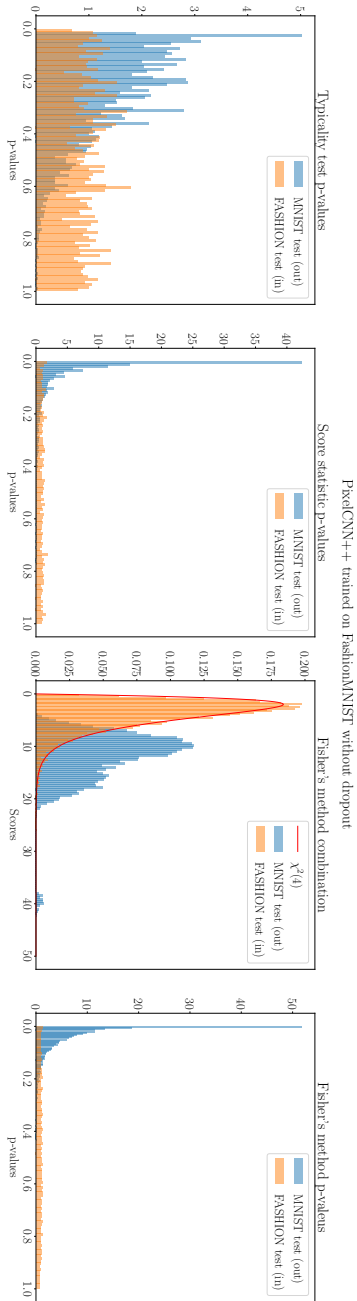
Figure 5.1: *First plot:* p-values of the typicality test on the two test sets. We can see that under $\mathcal{H}_0$, they should be uniformly distributed. *Second plot:* p-values of the score statistic. *Third plot:* values obtained by the Fisher's method. In red, we plot the density function of a χ^2-distribution with 4 degrees. This shows that the statistics are independent. *Fourth plot:* p-values obtained of the combination. These plots refer to a PixelCNN++ trained on FashionMNIST without dropout.

Table 5.1: AUROC↑ for single-sample OOD detection. For Fisher's method we mean the combination of the typicality test and the test statistic. These are also combined using DoSE.

	FashionMNIST (in) / MNIST (out)					
	SINGLE STATISTICS				COMBINATION	
MODELS	$\log p(x)$	$\|\nabla \log p(x)\|_2$	TYPICALITY	SCORE STAT	FISHER'S METHOD	DoSE_{KDE}
PixelCNN++ (dropout)	0.0762	0.8709	0.8314	**0.8822**	**0.9369**	0.8822
PixelCNN++ (no dropout)	0.1048	**0.9532**	0.7575	0.9381	**0.9536**	0.9382
Glow (RMSProp)	0.1970	0.8904	0.4807	**0.9114**	0.8598	**0.8901**
Glow (Adam)	0.1223	0.7705	0.6987	**0.8745**	**0.8839**	0.8752
HVAE	0.2620	0.8714	0.4884	**0.9578**	0.9383	**0.9498**

	CIFAR10 (in) / SVHN (out)					
	SINGLE STATISTICS				COMBINATION	
MODELS	$\log p(x)$	$\|\nabla \log p(x)\|_2$	TYPICALITY	SCORE STAT	FISHER'S METHOD	DoSE_{KDE}
PixelCNN++ (model1)	0.1553	**0.8006**	0.6457	0.6407	**0.6826**	0.6571
PixelCNN++ (model2)	0.1567	**0.7923**	0.6498	0.7067	**0.7300**	0.7243
Glow (RMSProp)	0.0630	0.8585	**0.8651**	0.7940	**0.8683**	0.8510
Glow (Adam)	0.0627	0.7844	**0.8624**	0.7655	**0.8613**	0.8588
HVAE	0.0636	0.8067	**0.8679**	0.7335	**0.8603**	0.8179

	CIFAR10 (in) / CIFAR100 (out)					
	SINGLE STATISTICS				COMBINATION	
MODELS	$\log p(x)$	$\|\nabla \log p(x)\|_2$	TYPICALITY	SCORE STAT	FISHER'S METHOD	DoSE_{KDE}
PixelCNN++ (model1)	0.5153	0.5306	**0.5458**	0.5362	**0.5563**	0.5477
PixelCNN++ (model2)	0.5150	0.5230	**0.5455**	0.5325	**0.5543**	0.5453
Glow (RMSProp)	0.5206	0.5547	0.5507	**0.5801**	**0.5844**	0.5842
Glow (Adam)	0.5206	0.5593	0.5508	**0.5692**	**0.5775**	0.5767
HVAE	0.5340	0.5280	0.5493	**0.5798**	0.5879	**0.5941**

description of these methods and additional results testing MNIST against FashionMNIST and SVHN against CIFAR10 can be found in appendix D.4.6. We also extensively analyzed, focusing mostly in the influence of the preprocessing, the results on CIFAR10 vs CelebA [424] in appendix D.5. In appendix D.4.7, we also considered a Gaussian Mixture Model and a Probabilistic PCA as simple generative models.

Models To analyze the effect of model architecture choices and optimization choice, we also consider different versions of the same model that reaches a similar log-likelihood. We consider 5 different models for each dataset pair. On FashionMNIST, we consider two Glow models, one trained using Adam and one using RMSProp and two PixelCNN++, trained with and without dropout. For CIFAR10, we consider two different PixelCNN++, one trained by us (model1) and one using a checkpoint given by the repository we used[5] (model2), and two

---

[5] `https://github.com/pclucas14/pixel-cnn-pp`

Glow models (Adam and RMSProp). For both datasets, instead, we consider only one HVAE.

Baselines   We are mostly interested in testing our methods with other model-agnostic test statistics in the literature. Apart from using the plain likelihood as an OOD score, the only test statistic we are aware of that can be applied to any generative model without requiring any background model or OOD assumptions is the typicality test statistic of Nalisnick et al. [473]. We also considered the gradient norm, which in general seem to work well but fails in the case of SVHN vs CIFAR10 (see appendix D.4.6). In addition to that, we compare our methods to a model-agnostic version of DoSE by Morningstar et al. [467], where we used KDEs to combine the score statistic and the typicality test statistic.

Evaluation   We compare our methods with the baselines by computing the area under the receiver operating characteristic curve (AUROC) as done in previous works [254, 467, 552]. We also evaluate our methods in terms of False Discovery Rate (FDR) control Benjamini and Hochberg [43], i.e. the proportion of false positive among the rejected hypothesis. Note that both quantities need to know the true label (OOD or in-distribution) to be computed.

5.7   RESULTS

One-sample OOD   We first evaluate our proposed method in the single-sample OOD detection task. Results are summarized in table 5.1. We start by considering the OOD task on FashionMNIST against MNIST. Looking at the single statistics, we notice that the score statistic is the one that works the best and the combination of the typicality test and the score statistic usually improve the AUROC than the two standalone statistics. In addition to that, it is better than the combination of the two statistics by using a KDE. DoSE seems to perform better on Glow trained with RMSProp, where the typicality is failing.

On natural images, instead, we have a different trend. The typicality test is better than the score statistic overall. The gradient norm surprisingly performs well in the two dataset pairs, but it fails badly when the model is trained on SVHN (see appendix D.4.6). Regarding the combination of the two statistics, the Fisher's method is always better than DoSE, but in this setting, it improves over the best of the single statistics three out of five times. In the *near*-OOD task, we have that both our method and DoSE using our suggested statistics perform closely. We want to highlight that for this challenging task we get results that are comparable with those reported in Morningstar et al. [467], but by using two model-agnostic statistics instead of three model-specific ones. It can be noticed that the way we train our models has a strong influence on both the typicality test

and the score statistic, although the models get the same test log-likelihood. In appendix D.4.4, we also show that this can happen between different checkpoints of the same model.

In figure 5.1, we show that the p-values distributions for both the typicality and the score statistic are uniformly distributed under the null-hypothesis and that the combination under the null follows a χ^2 distribution with 4 degrees of freedom. This also supports the fact that the typicality test and the score statistic are independent.

Two-sample OOD   As Nalisnick et al. [473], we consider how these test statistics change when performing two-sample OOD detection. Results are summarized in table 5.2. As shown by Nalisnick et al. [473], the typicality improves, but also the score statistic gets better if we consider more samples. Combining those leads to an improvement of performance in terms of AUROC with almost all the models. When training on FashionMNIST, the model can almost perfectly distinguish between the in-distribution test set and the OOD test set. While the performance improves for the two *far*-OOD task, we have that the improvement is slightly less evident in the *near*-OOD task of CIFAR10 vs CIFAR100.

5.7.1 Practical OOD detection with FDR control

One of the advantages of framing the problem as multiple testing is that we have a well-defined procedure to decide on which hypotheses to reject while controlling the False Discovery Rate (FDR, [43]). Imagine we are interested in finding the outliers from the dataset given by the combination of the two test-sets, but we do not want to discard too many inliers, then we can use the Benjamini-Hochberg (BH) procedure [43] to decide a threshold and reject all hypothesis below that threshold. For a specific significance level α, the procedure guarantees that the FDR stays below that level. Therefore, we can guarantee that the rate of inliers that are classified as outliers is less than the chosen α.

We leverage the fact that when the null hypothesis is true and the p-values are independent, then the scores obtained by combining k different statistics are χ^2_{2k} distributed to compute the p-values. Alternatively, the procedure can be also applied to the p-values of a single test-statistic. Usually, it is better to use an FDR control when it is actually possible to make few false discoveries, i.e. when we have a strong statistic. Therefore, we expect the procedure to work well when the AUROC is good, for examples on models trained on FashionMNIST.

As can be seen in figure 5.2, we have that the Type I ratio line stays below the identity line, meaning that the BH correction is working. When deciding for a specific threshold α, we usually have to trade off between Type I and Type II error and in most cases the threshold to choose depends on the application domain.

Table 5.2: AUROC↑ for two-sample OOD detection using the usual considered model.

MODELS	TYPICALITY	SCORE STAT	FISHER'S METHOD	DoSE$_{KDE}$
	FASHIONMNIST (IN) / MNIST (OUT)			
PCNN++ (drop.)	0.9514	0.9828	**0.9934**	**0.9912**
PCNN++ (no drop)	0.9081	0.9853	**0.9916**	**0.9921**
GLOW (RMSProp)	0.6190	**0.9588**	0.9187	0.7201
GLOW (Adam)	0.8525	**0.9716**	**0.9708**	**0.9736**
HVAE	0.6634	**0.9881**	0.9837	0.9889
	CIFAR10 (IN) / SVHN (OUT)			
PCNN++ (m1)	0.7675	0.6555	**0.7800**	0.7046
PCNN++ (m2)	0.7720	0.7235	**0.8227**	0.7850
GLOW (RMSProp)	0.9497	0.8624	**0.9536**	0.9379
GLOW (Adam)	0.9480	0.8370	**0.9519**	0.9329
HVAE	**0.9623**	0.7754	0.9560	0.9133
	CIFAR10 (IN) / CIFAR100 (OUT)			
PCNN++ (m1)	0.5433	0.5450	**0.5540**	**0.5508**
PCNN++ (m2)	0.5435	0.5370	**0.5533**	0.5470
GLOW (RMSProp)	0.5550	**0.6211**	0.6165	**0.6233**
GLOW (Adam)	0.5558	0.6073	0.6083	**0.6117**
HVAE	0.5594	0.6188	**0.6218**	**0.6273**

Ideally, we would like to have a low Type I and a low Type II error rate, meaning that we are not considering a lot of in-distribution examples as OOD and at the same time considering a lot of outliers as in-distribution. Figure 5.2 shows that we can achieve this for low values of α. When training on CIFAR, instead, we are able to control the FDR only from a certain significance level (see appendix D.4.5). This is expected given that the AUROC is not as good as when testing on MNIST.

5.8 DISCUSSION AND CONCLUSIONS

In this paper we studied the task of out-of-distribution detection using deep generative models and a combination of multiple statistical tests. We tested our method using different state-of-the art DGMs on classic image benchmark for OOD detection. We found that combining the two statistic leads to a more robust score that in some cases is close to state-of-the-art model-specific scores that require more assumptions. We also noticed that both the model design choice and the optimization choices have an influence on the score we are computing.
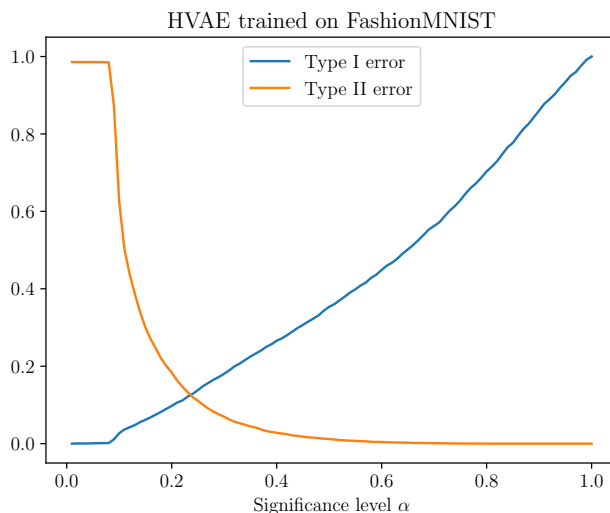
Figure 5.2: Type I (probability of an inlier to be classified as outlier) and Type II (probability of an outlier to be considered as inlier) errors versus the significance level α on the combination values. By using Benjamini-Hochberg correction, we get that the Type I error stays below identity line.

When considering only one-sided independent statistics, we showed that the Fisher's method tends to work better than combine them by summing the log-density of a KDE. We also noticed that the score statistic tends to perform a bit worse when the number of parameters of the models increases, i.e. in the context of natural images. One possible reason can be that in this setting the diagonal approximation is not good, and therefore one could consider different approximations, such as K-FAC.

DGMs have recently been used for handling missing data (see e.g. [293, 432, 447, 477]). An interesting future direction would be to extend these OOD detection methods to handle missing values.

The methods presented in this paper can also easily be applied when using model-specific one-sided statistics. In addition to obtain a more accurate score if one want to combine the test statistics, this also allows one to use well-defined procedure to control the FDR when choosing a which example to mark as outliers. Having this control, is necessary when we want to apply these methods in real settings.

UNSUPERVISED SPEECH REPRESENTATION LEARNING

A Brief Overview of Unsupervised Neural Speech Representation Learning

This chapter is a piece of original research by the candidate, previously published elsewhere and reprinted here with permission:

[C] Borgholt, L., **Havtorn, J. D.**, Edin, J., Maaløe, L., Igel, C., "A Brief Overview of Neural Speech Representation Learning". In: *Proceedings of the 2nd Workshop on Self-supervised Learning for Audio and Speech Processing (SAS) at the Thirty-Sixth AAAI Conference on Artificial Intelligence.* Virtual, 2022. arXiv: 2203.01829 [coauthor] [58]

ABSTRACT

Unsupervised representation learning for speech processing has matured greatly in the last few years. Work in computer vision and natural language processing has paved the way, but speech data offers unique challenges. As a result, methods from other domains rarely translate directly. We review the development of unsupervised representation learning for speech over the last decade. We identify two primary model categories: self-supervised methods and probabilistic latent variable models. We describe the models and develop a comprehensive taxonomy. Finally, we discuss and compare models from the two categories.

6.1 INTRODUCTION

Representation learning has shaped modern computer vision [606] and natural language processing [151], and more recently speech processing has been subject to the same development [26]. Representation learning has been defined as *"learning representations of the data that make it easier to extract useful information when building classifiers or other predictors"* [41]. Unsupervised representation learning is concerned with learning useful representations without the use of human annotations. Usually, a model is first pre-trained on a task where plenty of data is available. The model is then fine-tuned, or used to extract input representations for a smaller model, targeting a task with limited training data. In computer vision, both supervised [250, 606, 625] and unsupervised [158, 518] representation learning have gained attention with supervised representation learning driven by the availability of large annotated datasets [148]. For text and

speech, pre-training is usually unsupervised as labeled data is difficult to obtain. Although work on text has paved the way, and the two fields share many characteristics, learning representations from speech is a problem faced with a unique set of challenges.

In this paper, we survey work on unsupervised representation learning for speech processing from within the last decade. From a methodological perspective, we identify two primary model categories, namely models based on self-supervised learning and probabilistic latent variable models. We provide a methodological review of the design choices related to each of the model categories and develop a model taxonomy that highlights the different directions of work. Finally, we compare and discuss models from the two categories and their respective evaluation procedures.
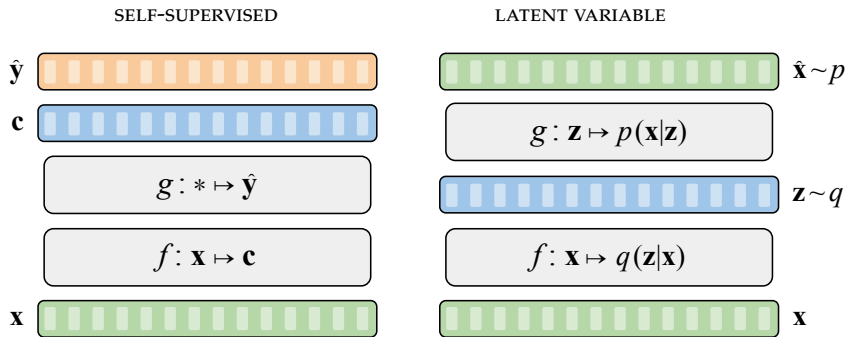
Figure 6.1: A schematic overview of the two groups of models covered in this survey. *Left*: A model trained with self-supervised learning. We take these models to consist of two functions $f(\cdot)$ and $g(\cdot)$ (section 6.2). After pre-training, $f(\cdot)$ is fine-tuned or used for extracting features **c**. $g(\cdot)$ is an auxiliary function used to accommodate the self-supervised pre-training task. *Right*: A probabilistic latent variable model. In contrast to the self-supervised model, the functions $f(\cdot)$ and $g(\cdot)$ learn the parameters of distributions q and p. The latent variable **z** is commonly used for representation learning.

6.2 UNSUPERVISED REPRESENTATION LEARNING

In the following, we group previous work into *self-supervised models* and *probabilistic latent variable models*, and take a *model* to comprise a neural architecture and a corresponding learning algorithm. A schematic overview is found in figure 6.1. These categories are neither exhaustive nor mutually exclusive, but allow us to focus on the characteristics that have shaped different branches of research.

With emphasis on recent successes in the field, we cover literature from the last 10 years. While a complete description of all relevant models is not within the scope of this work, we sketch important technicalities when they are particularly descriptive of certain models. We first define our high-level notation and conventions to ease discussion.

Notation   We use the subscript $i:j$ with $i \leqslant j$ to denote a vector sequence $\mathbf{a}_{i:j}$ containing elements $\mathbf{a}_i$ through $\mathbf{a}_j$. We denote model input as $\mathbf{x}_{1:T}$ which, in practice, might be either a spectrogram or the raw speech signal, but we do not distinguish between the two in notation as it is not essential to understand the models. Also, models commonly downsample the temporal dimension, but again, this is not crucial to understand the models, so we maintain a notation based on a single temporal dimension $t \in \{1, \ldots, T\}$.

When discussing self-supervised models, we use $\mathbf{c}_{1:T}$ to denote a contextualized representation. For stochastic latent variable models, we use $\mathbf{z}_{1:T}$ as is customary to the field. While some models are frozen and produce representations used as input for downstream tasks (**frz**, table 6.1), others are designed to be fine-tuned (**ftn**, table 6.1). In either case, we use $f(\cdot)$ to denote the model that is used for the downstream task. We use $g(\cdot)$ to denote any auxiliary model components (e.g., for a reconstruction task we might have $g : \mathbf{c}_t \mapsto \hat{\mathbf{x}}_t$). When a model can be naturally subdivided into multiple components, we simply use $f_*(\cdot)$ where $*$ may be any convenient descriptor. Finally, we often use a subscript when defining a loss, $\mathcal{L}_i$, to imply that the total loss is computed as a sum over i.

6.2.1   Self-supervised models

Self-supervised learning is a subset of unsupervised learning [649]. Where other unsupervised methods can be seen as a means to an end in itself (e.g., clustering or data generation), self-supervised learning takes the form of a pretext task that only adds value when associated with a downstream task. This makes self-supervised learning tie naturally with semi-supervised learning, but it may also be part of a fully unsupervised setup [23]. Self-supervised learning is often characterized by automatically deriving the target from the input or other unlabeled examples [505].

Predictive models   Similar to classic autoregressive language models [457], contextualized speech representations can be learned by predicting future values of a simple representation [114, 118, 311, 499, 584] (**prd**, table 6.1). Modeling spectrograms directly, autoregressive predictive coding (APC, [118]) is perhaps the

simplest example in this category. The forward pass and loss are computed as

$$\mathbf{c}_t = f(\mathbf{x}_{1:t}) \ , \tag{6.1}$$

$$\hat{\mathbf{x}}_{t+k} = g(\mathbf{c}_t) \ , \tag{6.2}$$

$$\mathcal{L}_t = \|\hat{\mathbf{x}}_{t+k} - \mathbf{x}_{t+k}\|_1 \ . \tag{6.3}$$

Here, $f(\cdot)$ and $g(\cdot)$ are parameterized by neural networks such that each $\mathbf{c}_t$ is only conditioned on previous inputs $\mathbf{x}_{1:t}$ and $\hat{\mathbf{x}}_{t+k}$ is computed step-wise. Chung et al. [118] use a stack of unidirectional LSTMs for $f(\cdot)$ and a linear regression layer for $g(\cdot)$. Tasks that seek to predict or reconstruct the input are very common. In the literature, these are often jointly referred to as reconstruction tasks (REC, table 6.1) [410, 679], although this is somewhat misleading in the case of prediction.

Contrary to generative models, such as WaveNet [496], the APC model is not restricted to next-step prediction. Instead, it predicts $k > 0$ steps ahead in order to ensure that the model does not learn a trivial solution by exploiting the smoothness of the signal. Depending on the downstream task, we are often interested in learning so-called slow features that will typically span multiple input frames [705]. Even the smallest linguistic units of speech, phonemes, tend to span 0.1 seconds on average [196], whereas spectrogram frames $\mathbf{x}_t$ are typically computed at 0.01 second intervals. However, sometimes local smoothness is explicitly used to define the task [20, 300, 301].

Contrastive models   Speech contains localized noise (e.g., phase shifts) that does not inform slow feature learning. Thus, directly modeling speech might not be the best way to learn contextualized representations. Contrastive predictive coding (CPC, [499]) targets a local variable $\mathbf{v}_{1:T}$, learned from the model input $\mathbf{x}_{1:T}$, instead of the input itself. The forward pass is

$$\mathbf{v}_t = f_v(\mathbf{x}_{t-r:t+r}) \ , \tag{6.4}$$

$$\mathbf{c}_t = f_c(\mathbf{v}_{1:t}) \ , \tag{6.5}$$

$$\hat{\mathbf{v}}_{t,k} = g_k(\mathbf{c}_t) \ , \tag{6.6}$$

where $f_v(\cdot)$ is a convolutional neural network, such that each $\mathbf{v}_t$ only encodes information from a limited receptive field $2r + 1$. Again, $f_c(\cdot)$ should be limited to condition each $\mathbf{c}_t$ on previous time-steps $\mathbf{v}_{1:t}$ and $g_k(\cdot)$ is a step-wise transformation. The loss is based on noise constrastive estimation [229] and is given by

$$\mathcal{L}_{t,k} = -\log\left(\frac{\exp(\hat{\mathbf{v}}_{t,k}^{\mathsf{T}}\mathbf{v}_{t+k})}{\sum_{n\sim\mathcal{D}}\exp(\hat{\mathbf{v}}_{t,k}^{\mathsf{T}}\mathbf{v}_n)}\right) \ . \tag{6.7}$$

Here, $\mathcal{D}$ is a set of indices including the target index $t + k$ and negative samples drawn from a proposal distribution, which is typically taken to be a uniform

distribution over the set $\{1, \ldots, T\}$. Note that the loss is also indexed by k to show that CPC targets multiple offsets. The APC model is easily extended in a similar way [115].

Crucially, we cannot simply predict $\mathbf{v}_{t+k}$ from $\mathbf{c}_t$ with an ℓ_1 loss. This would cause $f_v(\cdot)$ to collapse to a trivial solution, such as setting all $\mathbf{v}_t$ equal. With a contrastive loss on the other hand, setting all $\mathbf{v}_t$ equal would cause $\mathcal{L}_{k,t}$ to be constant at a value no better than a random baseline.

A model closely related to the original CPC model is wav2vec [584]. It uses a different parameterization of the functions $f_v(\cdot)$ and $f_c(\cdot)$, and modifies the loss to consider a binary prediction task, such that we have

$$\mathcal{L}_{t,k} = -\log(\sigma(\hat{\mathbf{v}}_{t,k}^\mathsf{T} \mathbf{v}_{t+k})) - \sum_{n \sim \mathcal{D}} \log(\sigma(-\hat{\mathbf{v}}_{t,k}^\mathsf{T} \mathbf{v}_n)). \tag{6.8}$$

This model was among the first to show that learned representations can be used to improve end-to-end speech recognition. As we will see, the wav2vec framework has evolved to shape state-of-the-art representation learning for speech.

Masking-based models   One downside of predictive tasks is that models are primarily unidirectional. Some work has extended APC and CPC inspired models with separate encoders operating in opposite directions [59, 331, 406], but these models are still restricted to process left and right context separately. Inspired by the masked language model task used for text-based representation learning [151], several papers have used masking to overcome this challenge (**msk**, table 6.1). Masking refers to replacing parts of the input with zeros or a learned masking vector. For zero-masking [101, 309, 405, 411, 690], we have

$$\mathbf{c}_{1:T} = f(\mathbf{x}_{1:T} \circ \mathbf{m}_{1:T}) \, , \tag{6.9}$$

$$\mathbf{x}_t = g(\mathbf{c}_t) \, , \tag{6.10}$$

$$\mathcal{L}_t = \|\mathbf{x}_t - \mathbf{x}_t\|_1 \, , \tag{6.11}$$

where the $\circ$ operator denotes the Hadamard product, $f(\cdot)$ is typically a transformer encoder or a bidirectional recurrent neural network, $g(\cdot)$ is a step-wise transformation, and $\mathbf{m}_{1:T}$ is a mask such that $m_{t,i} \in \{0, 1\}$. Alternatively, $\mathbf{m}_{1:T}$ is used to select which $\mathbf{x}_t$ are replaced by a learned masking vector. The entries of $\mathbf{m}_{1:T}$ are determined by some stochastic policy. One frequent inspiration is SpecAugment [507], which was originally proposed for supervised speech recognition and applies frequency and time masking to spectrogram representations. While temporal masking is most common, frequency masking has also been adopted for representation learning [690]. A simple, yet popular, masking strategy is to draw a proportion of input indices $t_i \sim \{1, \ldots, T - M\}$ without replacement, and then mask $\{t_i, \ldots, t_i + M\}$ [26, 275, 405].

Combining masking with a contrastive loss, wav2vec 2.0 was the first work to show that a competitive speech recognition model can be learned by fine-tuning a pre-trained model with as little as 10 minutes of labeled data. For this model

$$\mathbf{v}_t = f_\nu(\mathbf{x}_{t-r:t+r}) \ , \tag{6.12}$$

$$\mathbf{c}_{1:T} = f_c(\mathbf{v}_{1:T} \circ \mathbf{m}_{1:T}) \ , \tag{6.13}$$

$$\mathbf{q}_t = g_q(\mathbf{v}_t) \ . \tag{6.14}$$

Here, $f_\nu(\cdot)$ is a convolutional neural network, $f_c(\cdot)$ is a transformer encoder [663] and $g_q(\cdot)$ is a quantization module used to learn targets from the localized variable $\mathbf{v}_{1:T}$. Computing quantized targets this way requires an extra loss term, which we will present when we discuss quantization in general below. The contrastive loss for wav2vec 2.0 is similar to that of the CPC model,

$$\mathcal{L}_t = -\log\left(\frac{\exp(S_c(\mathbf{c}_t, \mathbf{q}_t))}{\sum_{n \sim \mathcal{D}} \exp(S_c(\mathbf{c}_t, \mathbf{q}_n))}\right) \ , \tag{6.15}$$

where $S_c(\cdot)$ is the cosine similarity and the negative samples in $\mathcal{D}$ are sampled from other masked time-steps.

In general, masking is less data efficient than prediction, as only the masked portion of the input is non-trivial to reconstruct. For this reason, the loss might be computed as

$$\mathcal{L}_t = \|(\mathbf{x}_t - \mathbf{x}_t) \circ (\mathbf{1} - \mathbf{m}_t)\|_1 \ . \tag{6.16}$$

Non-autoregressive predictive coding (NPC, [408]) tries to resolve this by using a convolutional neural network where the kernel is masked instead of the input. This allows for complete data utilization, but limits the amount of context encoded in the learned representation. figure 6.2 summarizes the models discussed so far.

Quantization   Several models enforce a discrete latent space by quantizing the vector representation (ǫтz, table 6.1). The two most popular approaches are the Gumbel-softmax [296, 440] and the quantization used in the VQ-VAE [500].

Gumbel-softmax approach: Say we want to quantize a vector $\mathbf{v}$ such that it takes one of K possible values. We first map $\mathbf{v}$ to $\mathbf{l} \in \mathbb{R}^K$ and then map $\mathbf{l}$ to a probability vector $\mathbf{p} \in \mathbb{R}^K$ via the Gumbel softmax given by

$$p_i = \frac{\exp(l_i + n_i)/\tau}{\sum_k^K \exp(l_k + n_k)/\tau} \tag{6.17}$$

for $i = 1, \ldots, K$. Here τ is a temperature parameter and $\mathbf{n} \in \mathbb{R}^K$ is a random vector with $n_i = -\log(-\log(u_i))$ for $u_i \sim \mathcal{U}(0, 1)$. As $\tau \to 0$, $\mathbf{p}$ approaches a

Table 6.1: Selected models classified according to the binary attributes identified throughout the text. The models are sorted according to first publication date on arXiv which might differ from the citation year. **MSK**: masking, **PRD**: prediction, **CON**: contrastive, **REC**: reconstruction, **QTZ**: quantization, **GEN**: generative, **FRZ**: frozen, **FTN**: fine-tuned, **LOC**: local, **GLO**: global.

MODEL	PUB. DATE	MSK	PRD	CON	REC	QTZ	GEN	LOC	GLB	VAR	FRZ	FTN
		MODEL AND TASK DESIGN						RESOLUTION			USAGE	
SELF-SUPERVISED MODELS												
Audio Word2vec [122]	2016 Mar.	✓	✗	✗	✓	✗	✗	✗	✓	✗	✓	✗
Speech2Vec [117]	2018 Mar.	✗	✓	✗	✓	✗	✗	✗	✓	✗	✓	✗
Unspeech [459]	2018 Apr.	✗	✓	✓	✗	✗	✗	✗	✓	✗	✓	✗
CPC [499]	2018 Jul.	✗	✓	✓	✗	✗	✗	✓	✗	✗	✓	✗
APC [118]	2019 Oct.	✗	✓	✗	✓	✗	✗	✓	✗	✗	✓	✗
wav2vec [584]	2019 Apr.	✗	✓	✓	✗	✗	✗	✓	✗	✗	✓	✗
Mockingjay [411]	2019 Oct.	✓	✗	✗	✓	✗	✗	✓	✗	✗	✓	✓
wav2vec 2.0 [26]	2020 Jun.	✓	✗	✓	✗	✓	✗	✓	✗	✗	✗	✓
NPC [408]	2020 Nov.	✓	✗	✗	✓	✓	✗	✓	✗	✗	✓	✗
DeCoAR 2.0 [405]	2020 Dec.	✓	✗	✗	✓	✓	✗	✓	✗	✗	✓	✗
SCPC [48]	2021 Jun.	✗	✓	✓	✗	✗	✗	✓	✗	✓	✓	✗
HuBERT [275]	2021 Jun.	✓	✗	✗	✗	✓	✗	✓	✗	✗	✗	✓
PROBABILISTIC LATENT VARIABLE MODELS												
VRNN [126]	2015 Jun.	✗	✗	✗	✓	✗	✓	✓	✗	✗	✓	✗
SRNN [188]	2016 May	✗	✗	✗	✓	✗	✓	✓	✗	✗	✓	✗
HMM-VAE [168]	2017 Mar.	✗	✗	✗	✓	✗	✓	✓	✗	✗	✓	✗
ConvVAE [277]	2017 Apr.	✗	✗	✗	✓	✗	✓	✗	✓	✗	✓	✗
FHVAE [278]	2017 Sep.	✗	✗	✗	✓	✗	✓	✓	✓	✗	✓	✗
VQ-VAE [500]	2017 Nov.	✗	✗	✗	✓	✓	✓	✓	✗	✗	✓	✗
BHMM-VAE [207]	2018 Mar.	✗	✗	✗	✓	✗	✓	✓	✗	✗	✓	✗
STCN [3]	2019 Feb.	✗	✗	✗	✓	✗	✓	✓	✗	✗	✓	✗
FDMM [335]	2019 Oct.	✗	✗	✗	✓	✗	✓	✓	✓	✗	✓	✗
ConvDMM [337]	2020 Jun.	✗	✗	✗	✓	✗	✓	✓	✗	✗	✓	✗

Figure 6.2: Schematic of self-supervised methods. Each subfigure illustrates the loss computation for a single time-step. The temporal subscript has been left out for simplicity.

one-hot vector. The Gumbel noise $\mathbf{n}$ is a practical way to sample from the untempered categorical distribution (i.e., $\tau = 1$). $\mathbf{p}$ is mapped to a one-hot vector using a function $\varphi(\cdot)$, such that $\varphi(\mathbf{p})_i = 1$ if $i = \arg\max_j p_j$ and 0 otherwise. As this function is non-differentiable, we must rely on the straight-through gradient estimator [42] which assumes that the Jacobian $\partial\varphi/\partial\mathbf{p}$ equals the identity matrix. The one-hot vector can then be used for a codebook lookup to obtain the final quantized vector (e.g., $\mathbf{q}_t$ in (6.14)).

The wav2vec 2.0 quantization module ((6.14)) uses the Gumbel softmax. To ensure utilization of codebook vectors, a diversity loss is added to the task specific loss ((6.15))

$$\mathcal{L} = -H(\widetilde{\mathbf{p}})/K \ , \tag{6.18}$$

where $H(\cdot)$ is the entropy and $\widetilde{\mathbf{p}}$ is the untempered version of $\mathbf{p}$ without Gumbel noise.

VQ-VAE approach: Instead of directly parameterizing a probability distribution, as in the Gumbel softmax, a vector $\mathbf{v}$ can be quantized by replacing it with the closest codebook vector $\mathbf{e}_k$. Specifically, given a learned codebook $\mathbf{e} \in \mathbb{R}^{K \times D}$,

where K is the codebook size and D is the dimensionality of each codebook vector $\mathbf{e}_k$, the quantized representation $\mathbf{q}$ of $\mathbf{v}$ is obtained as,

$$\mathbf{q} = \mathbf{e}_k \ , \quad \text{where} \quad k = \arg\min_j \left\| \mathbf{v} - \mathbf{e}_j \right\|_2 \ . \tag{6.19}$$

As $\arg\min$ is non-differentiable, the straight-through estimator is used as for the Gumbel-softmax. Codebook learning is facilitated by a two-term auxiliary loss similar to classical vector quantization dictionary learning [73, 617]. Gradients for the codebook vectors are given solely by a vector quantization term. A so-called commitment term ensures that non-quantized vectors do not grow unboundedly.

$$\mathcal{L} = \underbrace{\left\| \text{sg}\left[\mathbf{v}\right] - \mathbf{e} \right\|_2^2}_{\text{vq}} + \underbrace{\beta \left\| \mathbf{v} - \text{sg}\left[\mathbf{e}\right] \right\|_2^2}_{\text{commitment}} \ , \tag{6.20}$$

where $\text{sg}[\mathbf{x}] = \mathbf{x}$ is the stop-gradient operator with the property $\frac{d}{dx_i}\text{sg}[\mathbf{x}] \equiv 0$ for all i and β is a hyperparameter. Although vector quantization was introduced by the VQ-VAE which is, in some ways, a latent variable model, it has been applied to self-supervised methods [25, 661].

Motivation: Similar to how quantization approaches differ between works, so do the motivations provided for employing them. The vq-wav2vec [22, 25] learn quantized representations in order to apply natural language processing models, like BERT [151], afterwards. Other works use quantization for speech segmentation [110, 326] or as a bottleneck in order to "*limit model capacity*"[119, 405]. Finally, Chung, Tang, and Glass [119] explore quantization between different layers in the APC model, but find that continuous representations consistently perform better than their quantized counterparts on a downstream phoneme classification task

Given our previous discussion of how it might not be beneficial to model localized noise, quantization in wav2vec 2.0 seems well motivated, as it enforces the target representation $\mathbf{q}_{1:T}$ to discard such noise. Taking this idea further, the HuBERT model [275] uses offline quantization to learn categorical targets. Initially, spectrogram features are used to learn frame-wise labels with k-means clustering. A model similar to wav2vec 2.0, but without online quantization, is then trained to infer labels for masked time-steps. Since quantization is offline, this model does not need to rely on a contrastive loss, but can infer the target class directly. The offline quantization also ensures more stable training, as targets do not change abruptly.

Global representations The models covered so far learn representations that maintain a temporal resolution proportional to the input resolution. We say that

they learn local representations (LOC, table 6.1). Now, we cover models that learn global representations (GLB, table 6.1).

Early work on global speech representation learning takes inspiration from the autoencoder framework [352]. Chung et al. [122] propose a simple sequence-to-sequence autoencoder for learning acoustic word embeddings:

$$\mathbf{c} = f(\mathbf{x}_{1:T}) \ , \tag{6.21}$$

$$\hat{\mathbf{x}}_{1:T} = g(\mathbf{c}) \ , \tag{6.22}$$

where $f(\cdot)$ and $g(\cdot)$ are recurrent neural networks, such that $\mathbf{c}$ is taken to be the hidden state at the last time-step T of $f(\cdot)$ and used as initial hidden state of $g(\cdot)$. The authors also propose a denoising autoencoder with masked inputs $f(\mathbf{x}_{1:T} \circ \mathbf{m}_{1:T})$. Similar RNN-based autoencoders have also been explored [270, 321].

Prior to this work, Kamper et al. [322] and Renshaw et al. [555] introduced the *correspondence autoencoder*. This method uses dynamic time warping to align input-target segment pairs extracted with unsupervised term discovery. In more recent work, the need for alignment has been alleviated by adopting the sequence-to-sequence framework [295, 321].

Inspired by the work on semantic word embeddings for text [458], the sequence-to-sequence framework has also been used to implement speech-based versions of the skip-gram and continuous bag-of-words models [116, 117]. Given a segment corresponding to a single word $\mathbf{x}_{(n)} = \mathbf{x}_{t_n:t_{n+1}}$, the skip-gram model is trained to predict neighboring words $\mathbf{x}_{(n+k)}$ where $k \neq 0$. That is, instead of a single decoder, as in (6.21), the skip-gram model employs multiple decoders

$$\hat{\mathbf{x}}_{(n+k)} = g_k(\mathbf{c}) \ . \tag{6.23}$$

Conversely, the continuous bag-of-words model is trained to predict the target word from the neighboring words, so here multiple encoders sum over several offsets $\mathcal{K}$ to obtain $\mathbf{c}$:

$$\mathbf{c} = \sum_{k \in \mathcal{K}} f_k(\mathbf{x}_{(n+k)}) \ . \tag{6.24}$$

The sequence-to-sequence models described above rely on speech segments corresponding to words. The segments are obtained by supervised forced alignment, but similar models have been explored without this requirement [301, 628].

Contrastive learning has also been explored for global speech representation learning [298, 300, 459]. And prior to the widespread adoption of neural networks, Levin et al. [392] explore principal component analysis and Laplacian eigenmaps for learning fixed-sized acoustic embeddings.

Other work   Some models learn local representations with a variable temporal resolution that is not proportional to the input resolution (**var**, table 6.1). In practice, this is often achieved implicitly, by learning segment boundaries or by taking repeated quantized values to belong to the same segment [110, 154, 326, 354, 455, 683]. An exception is the recently proposed *segmental contrastive predictive coding* (SCPC, [48, 49]). With this approach, the model explicitly learns segment boundaries, which are used to downsample the representations during training. The same segmentation strategy has subsequently been applied in other models [137].

Most of the work presented so far fits neatly into the taxonomy presented in table 6.1. One exception is the problem-agnostic speech encoder (PASE, [516, 549]) that combines multiple pre-training tasks. Furthermore, many of the presented models have been successfully applied to other use cases. For instance, wav2vec 2.0 and related models have been applied to learn crosslingual and multilingual representations [131, 336, 558] and proven well-suited for concurrently learning with labeled data [630, 679].

6.2.2   Probabilistic latent variable models

Another prominent class of models are probabilistic latent variable models (LVMs). Before surveying their application to speech, we briefly review LVMs and their usual specification when applied for representation learning in general. We disregard any specific temporal notation without loss of generality. We then introduce the variational autoencoder framework (VAE, [340]). We focus on different dependency structures between data and learned representations, in contrast to the more practical view on self-supervised models taken above.

LVMs and inference   Fundamental to LVMs is the assumption that the data is produced by a generative process that involves unobserved stochastic latent variables $\mathbf{z}$. An LVM aims to model this generative process to enable generation of new data $\mathbf{x}$ (**gen**, table 6.1) and inference of the latent variable associated with a given observed variable $\mathbf{x}$. For representation learning, the inference of latent variables is of primary interest. An LVM is defined by the observation model $p(\mathbf{x}|\mathbf{z})$, which defines the relationship between the observed and latent variables, and the prior $p(\mathbf{z})$, which defines the relationship among the latent variables [33]. An LVM models the generative process via the joint observation and prior model $p(\mathbf{x}, \mathbf{z})$ often referred to as the generative model. The likelihood of an LVM given an example $\mathbf{x}$ can be written as

$$\log p(\mathbf{x}) = \log \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\, d\mathbf{z} \ . \tag{6.25}$$

The latent variable can be inferred with e.g. Markov Chain Monte Carlo (MCMC) methods [462] or variational inference [319].

For representation learning, LVMs are commonly defined using the VAE framework [340, 557] which is also the focus of our exposition. In the VAE framework, the observation model $p(\mathbf{x}|\mathbf{z})$ is parameterized using a deep neural network. This choice allows modeling complex and high-dimensional data but also makes the integral in (6.25) analytically intractable. MCMC methods can be used to estimate it and the true model posterior $p(\mathbf{z}|\mathbf{x})$, but these methods are usually computationally expensive in this setting [462]. To counter this and make gradient-based maximum likelihood training feasible, the VAE instead employs variational inference [319]. It approximates the intractable true model posterior by introducing a variational posterior distribution $q(\mathbf{z}|\mathbf{x})$, also parameterized by a deep neural network. From (6.25), via Jensen's inequality, this gives rise to a variational lower bound on the likelihood, also known as the evidence lower bound (ELBO).

$$\log p(\mathbf{x}) \geqslant \int q(\mathbf{z}|\mathbf{x}) \log \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x})} \, d\mathbf{z} \equiv \mathcal{L}_{\text{ELBO}} \ . \tag{6.26}$$

The bound can be efficiently evaluated and optimized with Monte Carlo (MC) estimation by sampling from $q(\mathbf{z}|\mathbf{x})$. Low-variance gradient estimates are usually obtained via reparameterization of $q(\mathbf{z}|\mathbf{x})$ [340], although alternatives exist (e.g., inverse CDF sampling) [462]. The ELBO can also be written as

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \left[\log p(\mathbf{x}|\mathbf{z}) \right] - D_{\text{KL}} \left(q(\mathbf{z}|\mathbf{x}) || p(\mathbf{z}) \right) \ , \tag{6.27}$$

where $\mathbb{E} \left[\log p(\mathbf{x}|\mathbf{z}) \right]$ can be seen as a reconstruction loss and $D_{\text{KL}} \left(q(\mathbf{z}|\mathbf{x}) || p(\mathbf{z}) \right)$ is the Kullback-Leibler (KL) divergence between the variational posterior distribution and the prior.

In brief, LVMs of the VAE type consist of an approximate posterior, $q(\mathbf{z}|\mathbf{x})$, an observation model, $p(\mathbf{x}|\mathbf{z})$, and a prior, $p(\mathbf{z})$. With reference to probabilistic coding theory, the approximate posterior is often referred to as the encoder and the observation model as the decoder [340, 557]. From a theoretical perspective, the encoder exists solely as the result of choosing to use variational inference to train the decoder rather than e.g. MCMC. As such, it is also referred to as the inference model. However, from a representation learning perspective, the encoder is essential as it can be used to efficiently obtain the representation $\mathbf{z}$ commonly used for downstream tasks. It is still possible to evaluate and sample the true posterior distribution $p(\mathbf{z}|\mathbf{x})$ by applying MCMC methods such as Hamiltonian Monte Carlo on the decoder, but for computational reasons this is rarely done in practice.

Table 6.2: A comprehensive overview of observation, prior and inference models for VAE type latent variable models with a single latent variable. The observation, prior and inference models may all belong to one or more of the categories listed under them as detailed in section 6.2.2. The types listed here serve as primitives from which more complex structures can be constructed including models with hierarchies of multiple latent variables.

Type		Form
Observation model		
arx	Autoregressive on $\mathbf{x}_t$	$p(\mathbf{x}_t|\mathbf{x}_{1:t-1})$
loc	Local latent variable	$p(\mathbf{x}_t|\mathbf{z}_{1:t})$
glb	Global latent variable	$p(\mathbf{x}_t|\mathbf{z})$
Prior		
arx	Autoregressive on $\mathbf{x}_t$	$p(\mathbf{z}_t|\mathbf{x}_{1:t-1})$
arz	Autoregressive on $\mathbf{z}_t$	$p(\mathbf{z}_t|\mathbf{z}_{1:t-1})$
ind	Locally independent $\mathbf{z}_t$	$p(\mathbf{z}_t)$
glb	Global latent variable	$p(\mathbf{z})$
Inference model		
arz	Autoregressive on $\mathbf{z}_t$	$q(\mathbf{z}_t|\mathbf{z}_{1:t-1})$
flt	Filtering	$q(\mathbf{z}_t|\mathbf{x}_{1:t})$
lsm	Local smoothing	$q(\mathbf{z}_t|\mathbf{x}_{t-r:t+r})$
gsm	Global smoothing	$q(\mathbf{z}_t|\mathbf{x}_{1:T})$
glb	Global latent variable	$q(\mathbf{z}|\mathbf{x}_{1:T})$

We next review VAEs applied to speech. We consider the choices of observation, prior and inference models. We provide a model taxonomy for selected LVMs in table 6.3.

Observation models A common choice for the observation model $p(\mathbf{x}|\mathbf{z})$ is to include an autoregressive dependency on the observed variable (**arx**, table 6.2) that is, $p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \cdot)$ where $\cdot$ represents some dependency on the latent variable [126, 188, 499, 500]. This allows the latent representation to focus on correlations that cannot easily be predicted from the observed variable at previous time-steps [499]. In practice, the dependency on $\mathbf{x}_{1:t-1}$ is often assumed to be Markovian and hence only on $\mathbf{x}_{t-1}$. Another common choice is to depend on a local window $\mathbf{x}_{t-r:t-1}$ where $r > 1$ is an integer denoting some receptive field. We will take a dependency on $\mathbf{x}_{1:t-1}$ to mean any one of these choices unless otherwise specified.

Table 6.3: Selected latent variable models classified according the attributes defined throughout section 6.2.2. See table 6.2 for the probability distributions that correspond to each of the attribute short-hands. HIE indicates a hierarchical representation.

MODEL	PUB. DATE	OBSERVATION			PRIOR				INFERENCE					HIE
		ARX	LOC	GLB	ARX	ARZ	IND	GLB	ARZ	FLT	LSM	GSM	GLB	
VRNN [126]	2015 Jun.	✓	✓	✗	✓	✓	✗	✗	✓	✓	✗	✗	✗	✗
SRNN [188]	2016 May	✓	✓	✗	✓	✓	✗	✗	✓	✗	✗	✓	✗	✗
HMM-VAE [168]	2017 Mar.	✗	✓	✗	✗	✓	✗	✗	✓	✓	✗	✗	✗	✓
ConvVAE [277]	2017 Apr.	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✓	✓	✗
FHVAE [278]	2017 Sep.	✗	✓	✓	✗	✗	✓	✓	✗	✗	✗	✓	✓	✓
VQ-VAE [500]	2017 Nov.	✓	✓	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗
BHMM-VAE [207]	2018 Mar.	✗	✓	✗	✗	✓	✗	✗	✓	✓	✗	✗	✗	✗
STCN [3]	2019 Feb.	✗	✓	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓
FDMM [335]	2019 Oct.	✗	✓	✓	✗	✓	✗	✓	✓	✓	✗	✗	✓	✓
ConvDMM [337]	2020 Jun.	✗	✓	✗	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗

While the autoregressive dependency might be important for learning a powerful generative model, it might not benefit the learned latent representations. Specifically encouraging the latent representation to discard correlations across the temporal dimension might degrade the quality of the latent representation. Furthermore, since such a decoder can perform quite well by simply solving an autoregressive prediction problem, similar to WaveNet [496], it can make the model prone to suffer from posterior collapse. This problem arises when the approximate and true posterior distributions collapse into the prior which renders the representations non-informative [61, 614]. Notably, posterior collapse is a local minimum of the ELBO since the KL-divergence becomes zero. Some works alleviate this problem with tricks like KL-annealing and free bits [61, 344, 614]. The VQ-VAE uses a quantized latent space that is not susceptible to posterior collapse *per se* [500]. How to equip LVMs with powerful decoders while avoiding posterior collapse is an open problem.

Some LVMs do not use autoregressive observation models [168, 207, 277, 278, 335, 337]. These more closely follow the assumption of *local independence* which states that observed variables are conditionally independent given the local (LOC, table 6.2) and/or global (GLB, table 6.2) latent variables [33]. However, this forces the latent variable to encode details about the observed variable to achieve a good reconstruction. This is opposite to contrastive self-supervised learning which allows models to discard details in $x_{1:T}$ that do not inform the training objective [26].

Priors   Priors can be said to belong to one or more of four broad categories. See table 6.2. Priors that are autoregressive on the observed variable (**ARX**, table 6.2) take the form $p(\mathbf{z}_t | \mathbf{x}_{1:t-1})$. This generally results in a slow-down of the generative process which may be of concern if the use-case is data generation. Priors that are autoregressive on the latent variable (**ARZ**, table 6.2) take the form $p(\mathbf{z}_t | \mathbf{z}_{1:t-1})$ and enable stochastic temporal transitions similar to hidden Markov models but with potentially nonlinear transition functions [126, 188, 335, 337]. Locally independent priors (**IND**, table 6.2) are rarely applied to sequential latent variables since they make the prior latent dynamics independent of the value of previous latent variables. Models that do impose such priors on sequential latents are quite limited in their generative power, unless they learn the prior dynamics post-hoc as done in the VQ-VAE [500]. Global latent variables (**GLB**, table 6.2) are fundamentally limited in the amount of information they can encode. Hence, models usually use them in combination with another local latent variable, or to encode fixed length input segments [277, 278, 335].

Inference models   LVMs based on the VAE perform so-called amortized variational inference. Here, a single inference network is used to infer the latent variables of any $\mathbf{x}$. For this reason, all inference models covered here are conditioned on the observed sequence in some way. Generally, the inference model can be seen as solving either a filtering or smoothing problem. In filtering (**FLT**, table 6.2), the latent variables are assumed to depend only on past and current values of the observed variable, $q(\mathbf{z}_t | \mathbf{x}_{1:t})$ [126, 337]. In global smoothing (**GSM**, table 6.2), this causal dependency is replaced with a dependency on all observed values, $q(\mathbf{z}_t | \mathbf{x}_{1:T})$ [188, 277]. Smoothing can also be done locally (**LSM**, table 6.2), where the latent variables then depend on $\mathbf{x}_{t-r:t+r}$ for some integer $r > 0$ [500]. Compared to self-supervised models that often use transformer encoders it can be hypothesized that global smoothing offers a stronger case than local smoothing and filtering for representation learning.

The inference model may also be used to infer a global latent variable (**GLB**, table 6.2) that might encode global information about $\mathbf{x}$. While it must be included in the prior model it might not be in the observation model, if the model also has a local latent variable. Finally, latent variables are often made to depend autoregressively on past inferred values, e.g. $q(\mathbf{z}_t | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})$ (**ARZ**, table 6.2) [126, 188].

Multiscale and hierarchical models   Some work has explored using a hierarchy of latent variables (**HIE**, table 6.3). This allows encoding the inductive bias that speech contains information at different temporal scales by letting the latent variables operate at different temporal scales [278]. Khurana et al. [335] propose using a temporal latent variable along with a global latent variable. Recent work has focused on learning a deeper latent hierarchy with five latent variables [3].

Other work   Before the introduction of the VAE, models such as deep belief networks (DBN, [262]) built from stacks of restricted Boltzmann machines [183, 610] were popular. Lee et al. [386] show the feasibility of using a two-layered DBN for discovering acoustic units of speech, while Deng et al. [149] show that a DBN can learn a binary coding of spectrograms that has higher signal-to-noise ratio than classical vector-quantization techniques for speech coding. DBNs are however notoriously tricky to optimize requiring the use of expensive MCMC sampling techniques for inference or resort to biased gradient estimates [182, 265]. Non-neural LVMs for speech representation learning have also been explored [251, 299, 383, 495].

6.3   DISCUSSION

From global to local   In table 6.1, we see that work on global representations within self-supervised learning precedes work on local representations. However, we find that the core ideas underlying the recent successes in learning local representation models have also been used for global representation learning; masking [122], context prediction [117], and contrastive training [459] have been applied in both settings. Furthermore, where work on global representation learning has taken inspiration from Word2vec [458], the techniques used for learning local representations are inspired by contextualized word embeddings [151]. Thus, the gap between these two model classes is largely a product of the developments in related fields and the general increase in computational resources.

Representations beyond inference   Predictive tasks are commonly used for self-supervised models, but they are not directly compatible with LVM training. However, an LVM prior with an autoregressive parameterization, $p(\mathbf{z}_t|\mathbf{z}_{1:t-1})$ or $p(\mathbf{z}_t|\mathbf{x}_{1:t-1})$, can be seen as predictive in the sense that it tries to match the approximate posterior. Hence, the prior might be considered for feature extraction. Jones and Moore [316] examine the importance of the prior in the VQ-VAE and show that the ability of this model to estimate densities $p(\mathbf{x}_{1:T})$ lies solely in its prior. Other work has also explored representations beyond the latent variable such as hidden units of the observation model [110, 337].

Masking and missing data   Masking may also improve representations learned with VAEs. Masking in VAEs has already been explored in the literature in the context of missing data imputation. Here, $\mathbf{x}$ is only partially observed, and often represented as a segmentation into observed and missing parts and a mask $\mathbf{m}$ indicating where the data is missing. The model is then trained to infer the latent variable from the observed data. Reconstruction also deals only with the

observed data. Previous work has largely focused on the ability of these models to yield high-quality imputations within the tabular and image data domains, without probing for the effects on the learned latent representation [293, 447]. The idea of using VAEs to impute missing data was already examined in the seminal paper by Rezende, Mohamed, and Wierstra [557]. Here the model was trained with fully observed data and used to impute data in an iterative sampling approach post hoc, leaving the learned representations unchanged.

Evaluating representations Although this review has a primarily methodological focus, we should briefly touch upon evaluation procedures. Training metrics for self-supervised tasks and the likelihood of LVMs offer little guidance as to the quality of the learned representations [288]. Thus, a common approach is to evaluate the representations in terms of their usefulness for downstream tasks. Such tasks may be chosen to target specific attributes of the representation (e.g. semantic or speaker information).

The SUPERB benchmark [723] gathers multiple tasks grouped into categories such as *recognition*, *detection*, *semantics*, *speaker*, *paralinguistics* and *generation*. The recently proposed SLUE benchmark focuses on spoken language understanding [603]. The long-standing zero resource speech challenge (ZeroSpeech) offers a new set of tasks for each edition [163, 164, 165, 166, 667] usually featuring a minimal-pair ABX task [579, 580].

Tasks that evaluate representations in terms of speaker-related information include speaker verification [278, 335, 459], speaker identification [118, 300, 408, 499], dialect classification [335], emotion recognition [516, 723] and gender classification [386]. The semantic content of representations are evaluated using tasks such as intent classification [465, 723], slot filling [361, 723], sentiment analysis [411], question answering [124], named entity recognition [56, 514, 603] and speech translation [29, 114]. Cardiac arrest detection for emergency calls has also been used to evaluate speech representations [56]. For local representations, phoneme classification is very common [110, 118, 277, 386, 410]. However, automatic speech recognition has become the *de facto* standard benchmark task [114, 275, 405].

Moving forward Most of the seminal work has focused on improving speech recognition [26, 584]. This focus has gained traction over the last couple of years, as computational resources have become more accessible and end-to-end models [86, 217] have been established as the dominant approach to speech recognition [223]. It is important to stress that self-supervised models, such as wav2vec 2.0 [26], represent a breakthrough, and recent successful approaches build upon this method. That is, deep self-attention models combined with masking [98, 275, 679]. This development mirrors years of rapid progress in masked language

modeling within natural language processing [129, 151], and we expect this to continue for unsupervised neural speech representation learning.

6.4   CONCLUSION

We reviewed unsupervised representation learning for speech, focusing on two primary categories: self-supervised methods and probabilistic latent variable models. Inspired by the development of self-supervised learning and the dependency structures of latent variable models, we derived a comprehensive model taxonomy. Finally, we compare and discuss models from the two categories and their respective evaluation procedures.

BENCHMARKING GENERATIVE LATENT VARIABLE MODELS FOR SPEECH

This chapter is a piece of original research by the candidate, previously published elsewhere and reprinted here with permission:

[D] **Havtorn, J. D.**, Borgholt, L., Hauberg, S., Frellsen, J., Maaløe, L., "Benchmarking Generative Latent Variable Models for Speech". In: *Proceedings of the Workshop on Deep Generative Models for Highly Structured Data at ICML*. 2022. arXiv: 2202.12707 [main author] [244]

7.1   ABSTRACT

Stochastic latent variable models (LVMs) achieve state-of-the-art performance on natural image generation but are still inferior to deterministic models on speech. In this paper, we develop a speech benchmark of popular temporal LVMs and compare them against state-of-the-art deterministic models. We report the likelihood, which is a much used metric in the image domain, but rarely, or incomparably, reported for speech models. To assess the quality of the learned representations, we also compare their usefulness for phoneme recognition. Finally, we adapt the Clockwork VAE, a state-of-the-art temporal LVM for video generation, to the speech domain. Despite being autoregressive only in latent space, we find that the Clockwork VAE can outperform previous LVMs and reduce the gap to deterministic models by using a hierarchy of latent variables.

7.2   INTRODUCTION

After the introduction of the variational autoencoder (VAE, Kingma and Welling [340] and Rezende, Mohamed, and Wierstra [557]) quickly came two temporal extensions for modeling speech data [126, 188]. Since then, temporal LVMs have undergone little development compared to their counterparts in the image domain, where LVMs recently showed superior performance to autoregressive models such as PixelCNN [497, 498, 573]. The improvements in the image domain have been driven mainly by top-down inference models and deeper latent hierarchies [102, 345, 434, 607, 614, 657]. In speech modeling however, autoregressive models such as the WaveNet remain state-of-the-art [496].

To compare and develop LVMs for speech, we need good benchmarks similar to those in the image domain. Image benchmarks commonly compare likelihood scores, but research in the speech domain often omits reporting a likelihood [278, 496, 500] or report likelihoods that are incomparable due to subtle differences in the assumed data distribution [3, 126, 188, 278]. Without a proper standard, it is difficult to compare explicit likelihood models for speech and develop them in an informed manner.

To advance the state of LVMs for speech, this paper (i) develops a benchmark for LVMs based on model likelihood, (ii) introduces a hierarchical LVM architecture without autoregressive decoder, (iii) compares LVMs to deterministic counterparts including WaveNet, and (iv) qualitatively and quantitatively evaluates the latent variables learned by different LVMs based on their usefulness for phoneme recognition. We find that:

(I) State-of-the-art LVMs achieve likelihoods that are inferior to WaveNet at high temporal resolution but are superior at lower resolutions. Interestingly, we find that a standard LSTM [269] almost matches the likelihood of WaveNet.

(II) LVMs with powerful autoregressive decoders achieve better likelihoods than the non-autoregressive LVM.

(III) The expressiveness of LVMs for speech increases with a deeper hierarchy of stochastic latent variables, similar to conclusions within image modeling.

(IV) LVMs learn rich representations that are as good or better than Mel spectrograms for phoneme recognition also when using only 10 minutes of labeled data.

At a high level, this benchmark brings order to LVM model comparisons for speech and also provides useful reference implementations of the models.[6] Before presenting the results, we provide a brief survey of existing LVMs for speech in a coherent notation.

7.3   LATENT VARIABLE MODELS FOR SPEECH

LVMs formulated as VAEs continue to be of interest since they are able to learn an approximation to the posterior distribution of assumed latent variables. The posterior is usually of a reduced dimensionality compared to the input and lies close to a known prior distribution. Approximate posteriors have various applications e.g. semi-supervised learning [343] and anomaly detection [245].

---

[6] `github.com/JakobHavtorn/benchmarking-lvms`
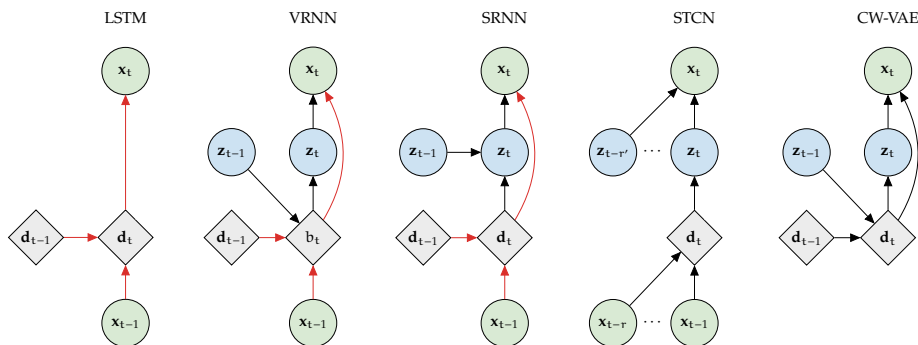
Figure 7.1: Generative models for a single timestep of a deterministic autoregressive LSTM, the VRNN and SRNN as well as the STCN and CW-VAE both with a single layer of latent variables. Red arrows indicate purely deterministic paths from the output x_t to previous input $x_{<t}$ without passing a stochastic node. The models differ in their use of latent variables and dependencies especially autoregressive dependencies and skip connections. We provide more elaborate graphical illustrations including inference models in figure 7.2 and appendix appendix C.10.

In recent years, several complementary methods have been proposed to improve the expressiveness of VAEs. These include building more expressive priors via methods such as normalizing flows [556] and building a deeper hierarchy of stochastic latent variables such as the Ladder VAE [614]. In this research, we focus on the latter due to the recent breakthroughs in image modeling using VAEs without costly autoregressive dependencies on the observed variable [102, 434, 657].

Several works have applied LVMs to speech. Among the first contributions were the VRNN [126] and SRNN [188] which can be seen as conditional VAEs per timestep. Other recent LVMs include the FH-VAE [278], which leverages an additional latent variable to capture global features, and Z-forcing [214], which resembles the SRNN but includes an auxiliary task in the latent space to increase its utilization. The VQ-VAE [500] is a hybrid between an LVM and an autoregressive model which uses a quantized latent space to improve the quality of generated samples. The Stochastic WaveNet [364] and STCN [3] use WaveNet encoder and decoders and temporally independent latent variables.

In this paper, we focus on the VRNN, SRNN and STCN. We exclude the Stochastic WaveNet as it is similar to STCN and achieves inferior likelihoods [3]. The FH-VAE, with disjoint latent variables and discriminative objective, Z-

forcing, with an auxiliary task, and the VQ-VAE, with a quantized latent space and autoregressive prior fitted after training, all introduce significant changes to the original VAE framework and are also not included here.

All selected models have autoregressive generative models which let future observed variables be generated by conditioning on previously generated values. Inspired by recent progress in the image domain, we therefore formulate and benchmark a novel temporal LVM which does not rely on an autoregressive decoder. We do so by adapting the hierarchical Clockwork Variational Autoencoder [577], originally proposed for video generation, to speech.

7.3.1 SEQUENTIAL DEEP LATENT VARIABLE MODELS

The selected models are all sequential deep latent variable models trained with variational inference and the reparameterization trick [340]. They take as input a variable-length sequence $\mathbf{x}_{1:T} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$ with $\mathbf{x}_t \in \mathbb{R}^{D_x}$. We let $\mathbf{x}_{1:T}$ refer to the observed variable or a downsampled version of it. We will sometimes use $\mathbf{x}$ to refer to the sequence $\mathbf{x}_{1:T}$ when it is not ambiguous.

First, $\mathbf{x}_{1:T}$ is encoded to a temporal stochastic latent representation $\mathbf{z}_{1:T} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T)$ with $\mathbf{z}_t \in \mathbb{R}^{D_z}$. This representation is then used to reconstruct the original input $\mathbf{x}_{1:T}$. The latent variable is assumed to follow some prior distribution $p(\mathbf{z}_t | \cdot)$ where the dot indicates that it may depend on latent and observed variables at previous timesteps, $\mathbf{z}_{<t}$ and $\mathbf{x}_{<t}$ where $\mathbf{z}_{<t} := (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{t-1})$.

The models are trained to maximize a likelihood objective. The exact marginal likelihood $\log p_\theta(\mathbf{x})$, where θ are parameters of the generative model, is intractable to optimize due to the integration over the latent space. Instead, we introduce the variational approximation $q_\phi(\mathbf{z}|\mathbf{x})$ to the true posterior. Via Jensen's inequality this yields the well-known evidence lower bound (ELBO) on the exact likelihood $\mathcal{L}(\theta, \phi; \mathbf{x})$ which can be jointly optimized with respect to $\{\theta, \phi\}$ using stochastic gradient descent methods. We omit the θ and ϕ subscripts for the remainder of the paper.

$$\log p_\theta(\mathbf{x}) = \log \int p_\theta(\mathbf{x}, \mathbf{z}) \, d\mathbf{z} \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] =: \mathcal{L}(\theta, \phi; \mathbf{x}) \ , \qquad (7.1)$$

Graphical illustrations of the models can be seen in figure 7.1 with more illustrations in appendix appendix C.10.

7.3.2 VARIATIONAL RECURRENT NEURAL NETWORK (VRNN)

The VRNN [126] is essentially a VAE per timestep. At timestep t, the VAE is conditioned on the hidden state of a recurrent neural network (RNN), $\mathbf{d}_t \in \mathbb{R}^{D_d}$, with state transition $\mathbf{d}_t = f([\mathbf{x}_{t-1}, \mathbf{z}_{t-1}], \mathbf{d}_{t-1})$ where $[\cdot, \cdot]$ denotes concatenation.

The VRNN uses a Gated Recurrent Unit (GRU, Cho et al. [105]) for f. The joint distribution factorizes over time and the latent variables are autoregressive in both the observed and latent space,

$$p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = \prod_{t=1}^{T} p(\mathbf{x}_t | \mathbf{x}_{<t}, \mathbf{z}_{\leqslant t}) p(\mathbf{z}_t | \mathbf{x}_{<t}, \mathbf{z}_{<t}) \ . \tag{7.2}$$

The approximate posterior similarly factorizes over time,

$$q(\mathbf{z}_{1:T} | \mathbf{x}_{1:T}) = \prod_{t=1}^{T} q(\mathbf{z}_t | \mathbf{x}_{\leqslant t}, \mathbf{z}_{<t}) \ . \tag{7.3}$$

From this, the ELBO for the VRNN is

$$\mathcal{L}(\mathbf{x}) = \mathbb{E}_{q(\mathbf{z}_{1:T} | \mathbf{x}_{1:T})} \left[\sum_t \log p(\mathbf{x}_t | \mathbf{x}_{<t}, \mathbf{z}_{\leqslant t}) - D_{KL} \left(q(\mathbf{z}_t | \mathbf{x}_{\leqslant t}, \mathbf{z}_{<t}) \ \| \ p(\mathbf{z}_t | \mathbf{x}_{<t}, \mathbf{z}_{<t}) \right) \right] \ . \tag{7.4}$$

The VRNN uses diagonal covariance Gaussian distributions $\mathcal{N}$ for the prior and posterior distributions. We denote the output distribution of choice by $\mathcal{D}$.

$$\begin{aligned} p(\mathbf{z}_t | \mathbf{x}_{<t}, \mathbf{z}_{<t}) &= \mathcal{N} \left(\alpha_p(\mathbf{d}_t) \right) \ , \\ p(\mathbf{x}_t | \mathbf{x}_{<t}, \mathbf{z}_{\leqslant t}) &= \mathcal{D} \left(\beta(\mathbf{z}_t, \mathbf{d}_t) \right) \ , \\ q(\mathbf{z}_t | \mathbf{x}_{\leqslant t}, \mathbf{z}_{<t}) &= \mathcal{N} \left(\alpha_q(\mathbf{x}_t, \mathbf{d}_t) \right) \ . \end{aligned} \tag{7.5}$$

All sets of distributional parameters, α_q, α_p and β, are the outputs of densely connected neural networks which we notationally overload as functions in equations (7.5) and section 7.3.2. It is common to refer to α_q as the inference model or encoder and β as the decoder. Together with β, the structural model α_p forms the generative model.

Since the decoder is dependent on $\mathbf{d}_t$, the transition function f allows the VRNN to learn to ignore parts of or the entire latent variable and establish a purely deterministic transition from $\mathbf{x}_{t-1}$ to $\mathbf{d}_t$ (figure 7.1). This failure mode is commonly referred to as posterior collapse and is a well-known weakness of VAEs with powerful decoders [61, 614].

7.3.3 STOCHASTIC RECURRENT NEURAL NETWORK (SRNN)

The SRNN [188] is similar to the VRNN but differs by separating the stochastic latent variables from the deterministic representations (figure 7.1). That is, the GRU state transition is independent of $\mathbf{z}_{1:T}$ such that $\mathbf{d}_t = f(\mathbf{x}_{t-1}, \mathbf{d}_{t-1})$. With this,

the joint $p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T})$ can be written as for the VRNN in (7.2). The approximate posterior of $\mathbf{z}_t$ is conditioned on the full observed sequence,

$$q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) = \prod_{t=1}^{T} q(\mathbf{z}_t|\mathbf{x}_{1:T}, \mathbf{z}_{t-1}) \ . \tag{7.6}$$

This is achieved by introducing a second GRU that runs backwards in time with transition $\mathbf{a}_t = g([\mathbf{x}_t, \mathbf{d}_t], \mathbf{a}_{t+1})$. While $p(\mathbf{x}_t|\mathbf{x}_{<t}, \mathbf{z}_{\leqslant t})$ remains as in (7.5), we have

$$\begin{aligned} p(\mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t}) &= \mathcal{N}\left(\alpha_p(\mathbf{z}_{t-1}, \mathbf{d}_t)\right) \ , \\ q(\mathbf{z}_t|\mathbf{x}_{1:T}, \mathbf{z}_{<t}) &= \mathcal{N}\left(\alpha_q(\mathbf{z}_{t-1}, \mathbf{a}_t)\right) \ . \end{aligned} \tag{7.7}$$

By inferring $\mathbf{z}_t$ conditioned on the full sequence $\mathbf{x}_{1:T}$, the SRNN performs smoothing. This has been noted to better approximate the true posterior of $\mathbf{z}_t$ which can be shown to depend on the full observed sequence [36]. Comparatively, the VRNN performs filtering.
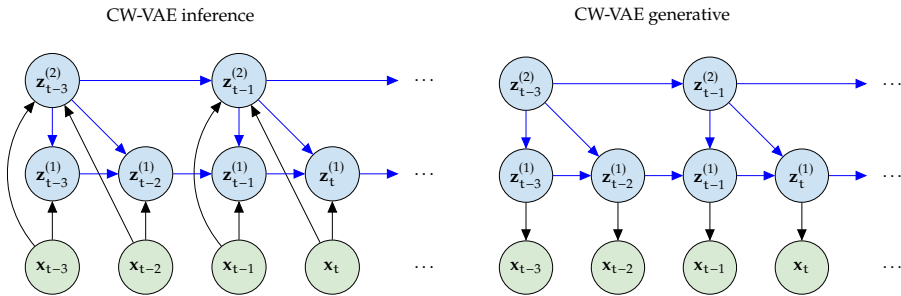
Figure 7.2: Inference (left) and generative (right) models for the CW-VAE with a hierarchy of $L = 2$ latent variables, $s_1 = 1$ and $s_2 = 2$. The models are unrolled over four consecutive timesteps but note that the graph continues towards $t = 0$ and $t = T$. Blue arrows indicate parameter sharing between inference and generative models. We omit the deterministic variable of figure 7.1 for clarity.

7.3.4   STOCHASTIC TEMPORAL CONVOLUTIONAL NETWORK (STCN)

The STCN [3] is a hierarchical latent variable model with an autoregressive generative model based on WaveNet [496]. Contrary to VRNN and SRNN, the latent variables have no transition functions connecting them over time. Instead, a latent $\mathbf{z}_t^{(l)}$ at layer l is conditioned on a window of observed variables $\mathbf{x}_{\mathcal{R}_t^{(l)}}$ via a WaveNet encoder where $\mathcal{R}_t^{(l)} = \{t - r_l + 1, \ldots, t\}$ and r_l is the receptive field of

the encoder at layer l. The window size grows exponentially with l.

$$p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}^{(1:L)}) = \prod_{t=1}^{T} p(\mathbf{x}_t | \mathbf{z}_{\mathcal{R}_t^{(1)}}^{(1:L)}) \prod_{l=1}^{L} p(\mathbf{z}_t^{(l)} | \mathbf{x}_{\mathcal{R}_{t-1}^{(l)}}, \mathbf{z}_t^{(l+1)}) \; ,$$

where $\mathbf{z}_t^{(L+1)} := \emptyset$ for notational convenience. The deterministic encoding is $\mathbf{d}_t^{(l)} = h(\mathbf{x}_{\mathcal{R}_t^{(l)}})$ where h is the encoder and $\mathbf{d}_t^{(l)}$ is extracted from the l'th layer similar to a Ladder VAE [614]. The approximate posterior is

$$q\left(\mathbf{z}_{1:T}^{(1:L)} | \mathbf{x}_{1:T}\right) = \prod_{t=1}^{T} \prod_{l=1}^{L} q\left(\mathbf{z}_t^{(l)} | \mathbf{x}_{\mathcal{R}_t^{(l)}}, \mathbf{z}_t^{(l+1)}\right) \; . \tag{7.8}$$

The factorized distributions are given as

$$p\left(\mathbf{z}_t^{(l)} | \mathbf{x}_{\mathcal{R}_{t-1}^{(l)}}, \mathbf{z}_t^{(l+1)}\right) = \mathcal{N}\left(\alpha_p^{(l)}\left(\mathbf{z}_t^{(l+1)}, \mathbf{d}_{t-1}^{(l)}\right)\right) \; ,$$

$$p\left(\mathbf{x}_t | \mathbf{z}_{\mathcal{R}_t^{(1)}}^{(1:L)}\right) = \mathcal{D}\left(\beta\left(\mathbf{z}_{\mathcal{R}_t^{(1)}}^{(1:L)}\right)\right) \; , \tag{7.9}$$

$$q\left(\mathbf{z}_t^{(l)} | \mathbf{x}_{\mathcal{R}_t^{(l)}}, \mathbf{z}_t^{(l+1)}\right) = \mathcal{N}\left(\alpha_q^{(l)}\left(\mathbf{z}_t^{(l+1)}, \mathbf{d}_t^{(l)}\right)\right) \; .$$

The decoder $\beta\left(\mathbf{z}_{\mathcal{R}_t^{(1)}}^{(1:L)}\right)$ is also a WaveNet.

7.3.5 CLOCKWORK VARIATIONAL AUTOENCODER (CW-VAE)

The CW-VAE [577] is a hierarchical latent variable model recently introduced for video generation. As illustrated in figures figure 7.1 and figure 7.2, it is autoregressive in the latent space but not in the observed space, contrary to the VRNN, SRNN and STCN. Additionally, each latent variable is updated only every s_l timesteps, where s_l is a layer-dependent integer, or stride, and $s_1 < s_2 < \cdots < s_L$. This imposes the inductive bias that latent variables exist at different temporal resolutions with $\mathbf{z}^{(l)}$ changing at lower frequency than $\mathbf{z}^{(l-1)}$. In speech, phonetic variation between $10 - 400$ ms, morphological and semantic features at the word level and speaker-related variation at the global scale make this a reasonable assumption.

To simplify temporal notation, we define the timesteps at which a layer updates its latent state as $\mathcal{T}_l := \{t \in [1, T] \,|\, (t-1) \bmod s_l = 0\}$. We then define the set of layers that update at a given timestep as $\mathcal{J}_t := \{l \,|\, t \in \mathcal{T}_l\}$. The joint distribution can then be written as,

$$p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}^{(1:L)}) = \prod_{t=1}^{T} p(\mathbf{x}_t | \mathbf{z}_t^{(1)}) \prod_{l \in \mathcal{J}_t} p(\mathbf{z}_t^{(l)} | \mathbf{z}_{t-1}^{(l)}, \mathbf{z}_t^{(l+1)}) \; .$$

The inference model is conditioned on a window of the observed variable $\mathbf{x}_{t:t+s_l}$ depending on the layer's stride s_l.

$$q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) = \prod_{t=1}^{T} \prod_{l \in \mathcal{J}_t} q\left(\mathbf{z}_t^{(l)}|\mathbf{z}_{t-1}^{(l)}, \mathbf{z}_t^{(l+1)}, \mathbf{x}_{t:t+s_l}\right) \ .$$

The dependency on $\mathbf{x}_{t:t+s_l}$ is encoded via a convolutional ladder network similar to the STCN with $\mathbf{d}_t^{(l)} = e(\mathbf{x}_{t:t+s_l})$. We define $\mathbf{x}_{\mathcal{S}_t^l} := \mathbf{x}_{t:t+s_l}$ for compactness. The latent state transitions are densely connected, and the decoder is also a convolutional network.

$$p\left(\mathbf{z}_t^{(l)}|\mathbf{z}_{t-1}^{(l)}, \mathbf{z}_t^{(l+1)}\right) = \mathcal{N}\left(\alpha_p^l\left(\mathbf{z}_{t-1}^{(l)}, \mathbf{z}_t^{(l+1)}\right)\right), \tag{7.10}$$

$$p\left(\mathbf{x}_t|\mathbf{z}_{t-s_l/2:t+s_l/2}^{(1)}\right) = \mathcal{D}\left(\beta\left(\mathbf{z}_{t-s_l/2:t+s_l/2}^{(1)}\right)\right),$$

$$q\left(\mathbf{z}_t^{(l)}|\mathbf{x}_{\mathcal{S}_t^l}, \mathbf{z}_{t-1}^{(l)}, \mathbf{z}_t^{(l+1)}\right) = \mathcal{N}\left(\alpha_q^l\left(\mathbf{z}_{t-1}^{(l)}, \mathbf{z}_t^{(l+1)}, \mathbf{d}_t^{(l)}\right)\right) \ .$$

7.3.6 Speech modeling with Clockwork VAEs

The video and speech modalities differ in the sampling rates normally used to digitize the natural signals. Sampling rates of common video codecs typically range from 24 Hz up to 60 or 120 Hz. In the speech domain, sampling rates range from 8000 Hz up to e.g. 44 100 Hz commonly used for music recordings. In the original work, s_l is defined as $s_l := k^{l-1}$ for some constant k which makes it exponentially dependent on the layer index l and forces $s_1 = 1$. While this is reasonable for the sample rates of video, training a model at this resolution is infeasible for audio waveform modeling. For this reason, we chose $s_1 \gg 1$ to achieve an initial temporal downsampling and let $s_l := c^{l-1}s_1$ for $l > 1$ and some constant c.

The encoder and decoder of the original CW-VAE are not applicable to speech. Hence, we parameterize them using 1D convolutions operating on the raw waveform. We use a ladder-network, similar to Aksan and Hilliges [3] and Sønderby et al. [614], for the encoder. A ladder-network leverages parameter sharing across the latent hierarchy and importantly processes the full observed sequence only once, sharing the resulting representations between latent variables. This yields a computationally efficient encoder and more activity in latent variables towards the top of the hierarchy.

7.3.7 Output distribution

Audio, as well as image data, are naturally continuous signals that are represented in discrete form in computers. The signals are sampled with some bit

depth b which defines the range of attainable values, $\mathbf{x} \in \{0, 1, \ldots, 2^b - 1\}$. The bit depth typically used in audio and image samplers is between 8 and 32 bit with 8 bit and 16 bit being the most common in the literature (MNIST [380], CelebA [424], CIFAR10 [356], TIMIT [196], LibriSpeech [506]).

In the image domain, the discrete nature of the data is usually modeled in one of two ways; either by using discrete distributions [102, 434, 573] or by dequantizing the data and using a continuous distribution [156, 267, 614], which yields a lower bound on the discrete distribution likelihood [636]. In the speech domain, however, the output distribution is often taken to be a continuous Gaussian [278, 364, 747] which was also originally done in VRNN, SRNN and STCN. This choice generally results in an ill-posed problem with a likelihood that is unbounded from above unless the variance is lower bounded [446]. As a result, *reported likelihoods can be sensitive to hyperparameter settings and be hard to compare*. We discuss this phenomenon further in appendix appendix C.8.

Most work normalizes the audio or standardizes it to values in a bounded interval $\mathbf{x} \in [-1, 1]$. Since $\mathbf{x}$ becomes approximately continuous as the bit depth b becomes large and the range of possible values becomes small, this alleviates the issue. However, commonly used datasets with bit-depths of 16 still result in a discretization gap between values that remains much larger than the gap between the almost continuous 32 bit floating point numbers which reinforces the problem [51].

In this work, we therefore benchmark models using a discretized mixture of logistics (DMoL) as output distribution. The DMoL was introduced for image modeling with autoregressive models [573] but has become standard in other generative models [102, 434, 657]. It was recently applied to autoregressive speech modeling of raw waveforms [501]. As opposed to e.g. a categorical distribution, the DMoL induces ordinality on the observed space such that values that are numerically close are also close in a probabilistic sense. This is a sensible inductive bias for images as well as audio where individual samples represent the amplitude of light or pressure, respectively. We discuss the DMoL for audio further in appendix appendix C.9.

7.4 Speech modeling benchmark

Data   We train models on TIMIT [196], LibriSpeech [506] and LibriLight [320]. For TIMIT, we randomly sample 5% of the training split for validation. We represent the audio as μ-law encoded PCM standardized to values in $[-1, 1]$ with discretization gap of 2^{-b+1}. We use the original bit depth of 16 bits and sample rate of 16 000 Hz. We use this representation both as the input and the reconstruction target. We provide more details on the datasets in appendix appendix C.3 and additional results on linear PCM in appendix appendix C.7.

Likelihood   We report likelihoods in units of bits per frame (bpf) as this yields a more interpretable and comparable likelihood than total likelihood in nats. It also has direct connections with information theory and compression [596, 648]. In units of bits per frame, lower is better. For LVMs, we report the one-sample ELBO. The likelihoods can be seen in tables 7.1 and 7.2. We describe how to convert likelihood to bpf in appendix appendix C.6.

Models   Architecture and training details are sketched below, while the full details are in appendices appendix C.4 and appendix C.5 along with additional results for some alternative model configurations in appendix appendix C.7. We select model configurations that can be trained on GPUs with a maximum of 12 GB of RAM and train all models until convergence on the validation set. We use a DMoL with 10 components for the output distribution of all models and model all datasets at their full bit depth of 16 bits. We train and evaluate models on stacked waveforms similar to previous work [3, 126, 188] with stack sizes of $s = 1$, $s = 64$ and $s = 256$. Hence, every model input $\mathbf{x}_t$ is composed of $\tilde{x}_{t:t+s}$ where $\tilde{x}$ are waveform frames. We provide additional results with a Gaussian output distribution in appendix appendix C.7.

   We configure WaveNet as in the original paper using ten layers per block and five blocks. We use $D_c = 96$ channels. We also train an LSTM model [269] which has fully connected encoders and decoders, as the VRNN and SRNN, but a deterministic recurrent cell and much fewer parameters than WaveNet. We report on LSTM models with $D_d = 256$ hidden units.

   The configuration of the VRNN and SRNN models is similar to the LSTM. For both models we set the latent variable equal in size to the hidden units, $D_z = 256$. At stack size $s = 1$, the models are computationally demanding and hence we train them on randomly sampled segments from each training example and only on TIMIT.

   The STCN is used in the "dense"configuration of the original work [3]. It uses 256 convolutional channels and $L = 5$ latent variables of dimensions 16, 32, 64, 128, 256 from the top down. We also run a one-layered ablation with the same architecture but only one latent variable of dimension 256 at the top. The CW-VAE is configured similar to the VRNN and SRNN models and with $c = 8$. We run the CW-VAE with $L = 1$ and 2 layers of latent variables. The number of convolutional channels and is set equal to D_z which is set to 96.

Baselines   We supply three elementary baselines that form approximate upper and lower bounds on the likelihood for arbitrary s. Specifically, we evaluate an uninformed per-frame discrete uniform distribution and a two-component DMoL fitted to the training set to benchmark worst case performance. We also report the likelihood achieved by the lossless compression algorithm, FLAC [130]

which establishes a notion of good performance, although not a strict best case. We report FLAC on linear PCM since it was designed for this encoding.

7.4.1 Likelihood results

Table 7.1: We report model likelihoods $\mathcal{L}$ for TIMIT represented as a 16 bit μ-law encoded PCM for different stochastic latent variable models and deterministic autoregressive baselines (left) and phoneme error rate (PER) of different representations for phoneme recognition on TIMIT (right).

s	Model	Configuration	$\mathcal{L}$ [bpf]		Data	Model	Input	PER [%]
1	Uniform	Uninformed	16.00					
1	DMoL	Optimal	15.60		3.7 h	Spectrogram	Mel	24.1
					3.7 h	WaveNet	$h^{(15)}$	27.7
1	WaveNet	$D_c = 96$	**10.88**		3.7 h	LSTM	h	23.0
1	LSTM	$D_d = 256$	10.97		3.7 h	VRNN	z	23.2
1	VRNN	$D_z = 256$	⩽11.09		3.7 h	SRNN	z	26.0
1	SRNN	$D_z = 256$	⩽11.19		3.7 h	CW-VAE	$z^{(1)}$	36.4
1	STCN	$D_z = 256, L = 5$	⩽11.77		3.7 h	STCN	$z^{(2)}$	**21.9**
64	WaveNet	$D_c = 96$	13.30		1.0 h	Spectrogram	Mel	30.8
64	LSTM	$D_d = 256$	13.34		1.0 h	WaveNet	$h^{(15)}$	34.7
64	VRNN	$D_z = 256$	⩽12.54		1.0 h	LSTM	h	30.1
64	SRNN	$D_z = 256$	⩽12.42		1.0 h	VRNN	z	30.4
64	CW-VAE	$D_z = 96, L = 1$	⩽12.44		1.0 h	SRNN	z	31.7
64	CW-VAE	$D_z = 96, L = 2$	⩽12.17		1.0 h	CW-VAE	$z^{(1)}$	40.0
64	STCN	$D_z = 256, L = 1$	⩽12.32		1.0 h	STCN	$z^{(2)}$	**26.7**
64	STCN	$D_z = 256, L = 5$	⩽**11.78**					
256	WaveNet	$D_c = 96$	14.11		10 m	Waveform	stacked	85.6
256	LSTM	$D_d = 256$	14.20		10 m	Spectrogram	Mel	47.1
256	VRNN	$D_z = 256$	⩽13.27		10 m	WaveNet	$h^{(15)}$	52.8
256	SRNN	$D_z = 256$	⩽13.14		10 m	LSTM	h	46.1
256	CW-VAE	$D_z = 96, L = 1$	⩽13.11		10 m	VRNN	z	44.6
256	CW-VAE	$D_z = 96, L = 2$	⩽12.97		10 m	SRNN	z	47.3
256	STCN	$D_z = 256, L = 1$	⩽13.07		10 m	CW-VAE	$z^{(1)}$	54.9
256	STCN	$D_z = 256, L = 5$	⩽**12.52**		10 m	STCN	$z^{(2)}$	**42.7**

TIMIT   For temporal resolutions of s = 1, the deterministic autoregressive models yield the best likelihoods with WaveNet achieving 10.88 bpf on TIMIT as seen in table 7.1 (left). Somewhat surprisingly, the LSTM baseline almost matches WaveNet with a likelihood of 11.11 bpf at s = 1. However, due to being autoregressive in training, the LSTM trains considerably slower than the parallel convolutional WaveNet; something not conveyed by table 7.1 (left). Notably, the VRNN and SRNN models achieve likelihoods close to that of WaveNet and the

Table 7.2: Model likelihoods $\mathcal{L}$ on LibriSpeech test sets represented as 16 bit μ-law encoded PCM. For the CW-VAE, s refers to s_1 and the two-layered models have $s_2 = 8s_1$. The models are trained on either the 10 h LibriLight subset or the 100 h LibriSpeech train-clean-100 subset as indicated. Likelihoods are given in units of bits per frame (bpf).

s	Model	Configuration	Likelihood $\mathcal{L}$ [bpf]			
			dev-clean 10h/100h	dev-other 10h/100h	test-clean 10h/100h	test-other 10h/100h
1	Uniform	Uninformed	16.00	16.00	16.00	16.00
1	DMoL	Optimal	15.66	15.70	15.62	15.71
1	WaveNet	$D_c = 96$	**10.96/10.89**	**10.85/10.76**	**11.12/11.01**	**11.05/10.85**
1	LSTM	$D_d = 256$	11.21/11.17	11.10/11.06	11.35/11.29	11.28/11.23
64	WaveNet	$D_c = 96$	13.61/13.24	13.58/13.21	13.61/13.22	13.60/13.21
64	LSTM	$D_d = 256$	13.56/13.25	13.52/13.24	13.55/13.23	13.56/13.25
64	CW-VAE	$D_z = 96, L = 1$	⩽12.32/12.24	12.32/12.23	12.43/12.33	12.43/12.33
64	CW-VAE	$D_z = 96, L = 2$	⩽12.30/12.22	12.30/12.21	12.40/12.31	12.39/12.32
64	STCN-dense	$D_z = 256, L = 5$	⩽**11.98/11.47**	**11.98/11.46**	**12.08/11.58**	**12.09/11.60**
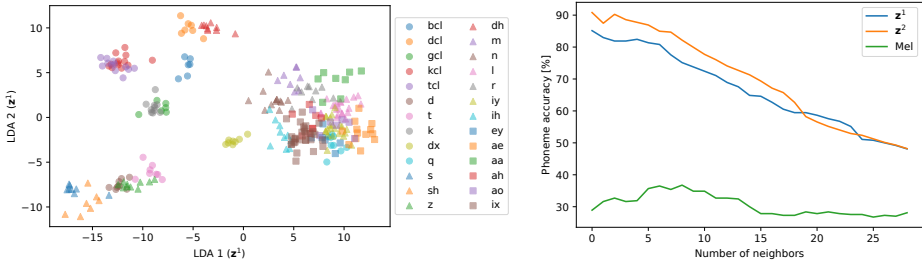
Figure 7.3: (left) Clustering of phonemes in a 2D Linear Discriminant Analysis (LDA) subspace of a CW-VAE latent space ($\mathbf{z}^{(1)}$). (right) Leave-one-out phoneme classification accuracy for a KNN classifier at different K in a 5D LDA subspace of a CW-VAE latent space.

LSTM at around 11.09 bpf. The STCN exhibited instability when trained at s = 1 and tended to diverge.

At s = 64, WaveNet and the LSTM yield significantly worse likelihoods than all LVMs separated by ~ 1 bit. The CW-VAE outperforms the VRNN and SRNN when configured with a hierarchy of latent variables. With a single layer of latent variables, the CW-VAE is inferior to both SRNN and VRNN but notably still better than the LSTM. These observations carry to s = 256, where a multilayered CW-VAE outperforms the LSTM, VRNN and SRNN. The STCN yields the best results at both s = 64 and s = 256. For strides s > 1, previous work has attributed

the inferior performance of autoregressive models without latent variables, such as WaveNet and the LSTM, to the ability of LVMs to model intrastep correlations [363].

Decreasing the resolution s improves the likelihood for all LVMs. However, the best performing models, STCN and CW-VAE are not yet scalable to this regime for reasons related to numerical instability and computational infeasibility. This seems to indicate that LVMs may be able to outperform autoregressive models at $s = 1$ in the future.

LibriSpeech   On LibriSpeech (table 7.2), results are similar to TIMIT. The STCN achieves the best likelihood at $s = 64$ and the CW-VAE surpasses WaveNet and the LSTM.

Compression   A connection can be made between the model likelihoods and the compression rates of audio compression algorithms. Lossy compression algorithms, such as MP3, exploit the dynamic range of human hearing to achieve 70-95% reduction in bit rate [62] while lossless compression algorithms, such as FLAC, achieve 50-70% reduction [130] independent of audio content. Although both the autoregressive models and the LVMs are lossy, their objective minimizes the amount of incurred loss. The best likelihoods reported in table 7.1 (left) and table 7.2 correspond to about a 30% reduction in bit rate which indicates that there are significant gains in likelihood to be made in speech modeling.

7.5   PHONEME RECOGNITION

Although the likelihood is a practical metric for model comparison and selection, a high likelihood does not guarantee that a model has learned useful representations [288]. For speech data, we would expect models to learn features related to phonetics which would make them useful for tasks such as automatic speech recognition (ASR). The Mel spectrogram is a well-established representation of audio designed for speech recognition and is predefined rather than learned. To assess the usefulness of the representations learned by the benchmarked models, we compare them to the highly useful Mel spectrogram on the task of phoneme recognition. Phonemes are fundamental units of speech that relate to how parts of words are pronounced (see also appendix appendix C.12).

Quantitative   We train an ASR model to recognize phonemes and compare its performance when using input representations obtained from different unsupervised models. For WaveNet and the LSTM, we use the hidden state as the representation. For all LVMs, we use the latent variable. For the hierarchical LVMs

and WaveNet, we run the experiment using each possible representation and report only the best one here. We compare the learned representations to a log Mel spectrogram with 80 filter banks, hop length 64 and window size 128. We also compared to using raw PCM in vectors of 64 elements standardized to $[-1, 1]$ but found that the ASR did not reliably converge at all which highlights the importance of input representation. The ASR model is a three-layered bidirectional LSTM with 256 hidden units. It is trained with the connectionist temporal classification (CTC) loss [217] which lets it jointly learn to align and classify without using label alignments. We pre-train all unsupervised models at $s = 64$ on the full TIMIT training dataset excluding the validation data (3.7h) as in table 7.1 (left). We then train the ASR model on all 3.7 h as well as 1 h and 10 m subsets. We report results on the test set in terms of phoneme error rates (PER) in table 7.1 (right).

As expected, Mel spectrogram performs well achieving 24.1% PER using 3.7 hours of labeled data. However, the ASR trained on STCN representations outperforms the Mel spectrogram with a PER of 21.9%. This indicates that unsupervised STCN representations are phonetically rich and potentially better suited for speech modeling than the engineered Mel spectrogram. When the amount of labeled data is reduced, LVM representations suffer slightly less than deterministic ones. WaveNet representations are interestingly outperformed by both the LSTM and all LVMs.

Qualitative We qualitatively assess the learned latent representations for the CW-VAE. We infer the latent variables of all utterances by a single speaker from the TIMIT test set. We sample the latent sequence 100 times to estimate the mean representation per timestep. We then compute the average latent representation over the duration of each phoneme using aligned phoneme labels. This approximately marginalizes variation during the phoneme. We use linear discriminant analysis (LDA) [185] to obtain a low-dimensional linear subspace of the latent space. We visualize the resulting representations colored according to test set phoneme classes in figure 7.3. We note that many phonemes are separable in the linear subspace and that related phonemes such as "sänd "shäre close.

We also show the average accuracy of a leave-one-out k-nearest-neighbor (KNN) classifier on the single left-out latent representation reduced with a 5-dimensional LDA as a function of the number of neighbors. We compare accuracy to a Mel-spectrogram averaged over each phoneme duration and LDA reduced. The spectrogram is computed with hop length set to 64, equal to s_1 for the CW-VAE, window size 256 and 80 Mel bins. We see that both latent spaces yield significantly better KNN accuracies than the Mel features.

7.6 CONCLUSION

In this paper, we developed a benchmark for speech modeling with stochastic latent variable models (LVMs). We compared LVMs and deterministic autoregressive models on equal footing and found that LVMs achieve inferior likelihood compared to deterministic WaveNet and LSTM baselines. Surprisingly, the LSTM almost matched the popular WaveNet model. We saw that hierarchical LVMs, such as STCN and CW-VAE, outperformed non-hierarchical versions of themselves in ablation experiments as well as non-hierarchical LVMs such as VRNN and SRNN. This matches recent observations in the image domain. We noted that the STCN with an autoregressive decoder outperforms the non-autoregressive CW-VAE, which we adapted to speech. Finally, we found that LVMs can learn latent representations that are useful for phoneme recognition and better than Mel spectrograms, which are tailored for the task, when identical models are trained on top of the representations. While the best performing models are not yet scalable to the highest temporal resolution, these results indicate that they might improve upon deterministic models in the future.

PART IV

MEDICAL APPLICATIONS

Automated Medical Coding on MIMIC-III and MIMIC-IV: A Critical Review and Replicability Study

This chapter is a piece of original research by the candidate, previously published elsewhere and reprinted here with permission:

[E] Edin, J., Junge, A., **Havtorn, J. D.**, Borgholt, L., Maistro, M., Ruotsalo, T., Maaløe, L., "Automated Medical Coding on MIMIC-III and MIMIC-IV: A Critical Review and Replicability Study". In: *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. Taipei, Taiwan: ACM, 2023. arXiv: 2304. 10909 [coauthor] [171]

Abstract

Medical coding is the task of assigning medical codes to clinical free-text documentation. Healthcare professionals manually assign such codes to track patient diagnoses and treatments. Automated medical coding can considerably alleviate this administrative burden. In this paper, we reproduce, compare, and analyze state-of-the-art automated medical coding machine learning models. We show that several models underperform due to weak configurations, poorly sampled train-test splits, and insufficient evaluation. In previous work, the macro F1-score has been calculated suboptimally, and our correction doubles it. We contribute a revised model comparison using stratified sampling and identical experimental setups, including hyperparameters and decision boundary tuning. We analyze prediction errors to validate and falsify assumptions of previous works. The analysis confirms that all models struggle with rare codes, while long documents only have a negligible impact. Finally, we present the first comprehensive results on the newly released MIMIC-IV dataset using the reproduced models. We release our code, model parameters, and new MIMIC-III and MIMIC-IV training and evaluation pipelines to accommodate fair future comparisons.[7]

---

[7]https://github.com/JoakimEdin/medical-coding-reproducibility

8.1 INTRODUCTION

Medical coding is the task of assigning diagnosis and procedure codes to free-text medical documentation [159, 633]. These codes ensure that patients receive the correct level of care and that healthcare providers are accurately compensated for their services. However, this is a costly manual process prone to error [72, 492, 650].

The goal of automated medical coding (AMC) is to predict a set of codes or provide a list of codes ranked by relevance for a medical document. Numerous machine learning models have been developed for AMC [308, 620, 633]. These models are trained on datasets of medical documents, typically discharge summaries, each labeled with a set of medical codes. While some models treat AMC as an ad-hoc information retrieval problem [509, 559], it is more commonly posed as a multi-label classification problem [308, 633].

While most research in AMC has been conducted on the third version of the Medical Information Mart for Intensive Care dataset (MIMIC-III) [633, 666], it remains challenging to compare the results of different models. Performance improvements are commonly attributed to model design, but differences in experimental setups make these claims hard to validate. In addition, long documents, rare codes, and lack of training data are often cited as core research challenges [30, 159, 160, 181, 194, 281, 307, 308, 330, 338, 398, 419, 454, 464, 515, 633, 634, 666, 671, 717, 724, 740, 744]. However, except for a few studies demonstrating performance drops on rare codes, the number of studies containing in-depth error analyses is limited [30, 160, 307].

We address the above challenges. Our major contributions are:

1. We reproduce the performance of state-of-the-art models on MIMIC-III. We find that evaluation methods are flawed and propose corrections that double the macro F1-scores.

2. We find the original split of MIMIC-III to introduce strong biases in results due to missing classes in the test set. We create a new split with full class representation using stratified sampling.

3. We perform a revised model comparison on MIMIC-III *clean* using the same training, evaluation, and experimental setup for all models. We find that models previously reported as low-performing improve considerably, demonstrating the importance of hyperparameters and decision boundary tuning.

4. We report the first results of current state-of-the-art models on the newly released MIMIC-IV dataset [209, 314]. We find that previous conclusions generalize to the new dataset.

5. Through error analysis, we provide empirical evidence for multiple model weaknesses. Most importantly, we find that rare codes harm performance, while, in contrast to previous claims, long documents only have a negligible performance impact.

We release our source code and new splits for MIMIC-III and IV[1] and hope these contributions will aid future research in AMC.

8.2 Previous work

In the following, we review datasets, model architectures, training, and evaluation of the models we compare in this study. Our criteria for selecting these models are presented in section 8.3.1.

8.2.1 Datasets

The International Classification of Diseases (ICD) is the most popular medical coding system worldwide [633]. It follows a tree-like hierarchical structure, also known as a medical ontology, to ensure the functional and structural integrity of the classification. Chapters are the highest level in the hierarchy, followed by categories, sub-categories, and codes. The World Health Organization (WHO) revises ICD periodically. Each revision introduces new codes. For instance, ICD-9 contains 18,000 codes, while ICD-10 contains 142,000.[8] MIMIC-II and MIMIC-III are the most widely used open-access datasets for research on ICD coding and are provided by the Beth Israel Deaconess Medical Center [315, 387, 633].

MIMIC-III contains medical documents annotated with ICD-9 codes collected between 2001 and 2012 [315]. Usually, discharge summaries—free-text notes on patient and hospitalization history—are the only documents used for AMC [633]. MIMIC-III *full* and *50* are commonly used splits. MIMIC-III *full* contains all ICD-9 codes, while *50* only contains the top 50 most frequent codes [469, 600].

MIMIC-IV was released on January 6th, 2023, and has not previously been used for AMC. It contains data for patients admitted to the Beth Israel Deaconess Medical Center emergency department or ICU between 2008-2019 annotated with either ICD-9 or ICD-10 codes [314]. The empirical frequencies of codes of each ICD version in MIMIC-IV are shown in figure 8.1. Statistics for the MIMIC-III *50*, *full*, and MIMIC-IV datasets are listed in table 8.1.

---

[8]https://www.cdc.gov/nchs/icd/icd10cm_pcs_background.htm

Table 8.1: Comparison of the previously defined MIMIC-III splits *full* and 50 [469] and our proposed MIMIC-III *clean* split along with similarly defined splits for MIMIC-IV *ICD-9* and *ICD-10* after pre-processing.

	Previous work		Our work		
	MIMIC-III *full*	MIMIC-III 50	MIMIC-III *clean*	MIMIC-IV *ICD-9*	MIMIC-IV *ICD-10*
Number of documents	52,723	11,368	52,712	209,326	122,279
Number of patients	41,126	10,356	41,118	97,709	65,659
Number of unique codes	8,929	50	3,681	6,150	7,942
Codes pr. instance: Median (IQR)	14 (10-20)	5 (3-8)	14 (10-20)	12 (8-17)	14 (9-20)
Words pr. document: Median (IQR)	1,375 (965-1,900)	1,478 (1,065-1,992)	1,311 (917-1,822)	1,320 (997-1,715)	1,492 (1,147-1,931)
Documents: Train/val/test [%]	90.5/3.1/6.4	71.0/13.8/15.2	72.9/10.6/16.6	73.8/10.5/15.7	72.9/10.9/16.2
Missing codes: Train/val/test [%]	2.7/66.4/54.3	0.0/0.0/0.0	0.0/0.1/0.0	0.0/0.5/0.2	0.0/0.5/0.1
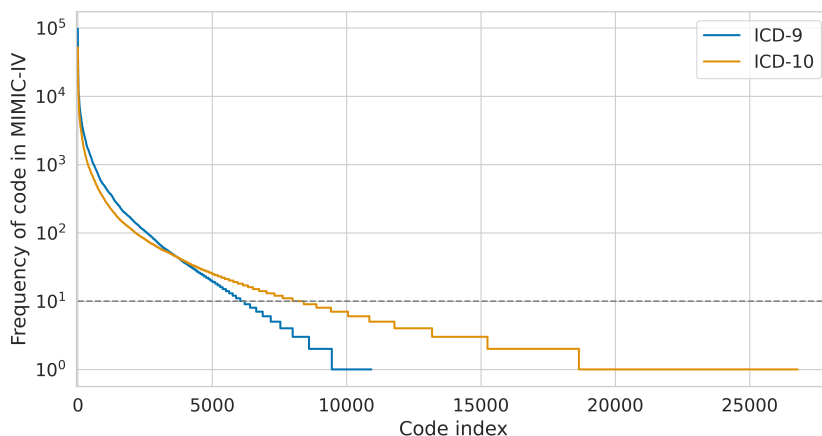
Figure 8.1: The frequency of ICD-9 and ICD-10 codes in MIMIC-IV before pre-processing. As discussed in section 8.3.3, we removed codes with fewer than ten instances (dashed line).

8.2.2   MODEL ARCHITECTURES

Most recent state-of-the-art AMC models use an encoder-decoder architecture. The encoder takes a sequence of tokens $T \in \mathbb{Z}^n$ as input and outputs a sequence of hidden representations $H \in \mathbb{R}^{d_h \times n}$, where n is the number of tokens in a sequence, and d_h is the hidden dimension. The decoder takes H as input and outputs the code probability distributions. For the task of ranking, codes are sorted by decreasing probability. For classification, code probabilities larger than a set decision boundary are predicted.

Encoders: The encoder usually consists of pre-trained non-contextualized word embeddings (e.g., Word2Vec) and a neural network for encoding context. More recently, pre-trained masked language models (e.g., BERT) have gained popularity [633]. The MIMIC-III training set or PubMed articles are commonly used for pre-training.

Decoders: The most common decoder architectures can be grouped into three primary types. The simplest decoder is a pooling layer (e.g., max pooling) followed by a feed-forward neural network. More recently, label-wise attention (LA) [469] has replaced pooling [281, 398, 419, 671]. LA transforms a sequence of hidden representations H into label-specific representations $V \in \mathbb{R}^{d_h \times L}$, where

L is the number of unique medical codes in the dataset. It is computed as

$$A = \text{softmax}(WH), \quad V = HA^{\top} , \tag{8.1}$$

where the softmax normalizes each column of WH, $W \in \mathbb{R}^{L \times d_h}$ is an embedding matrix that learns label-specific queries, and $A \in \mathbb{R}^{L \times n}$ is the attention matrix. Then, V is used to compute class-wise probabilities via a feedforward neural network. As LA was first used in the *convolutional attention for multi-label classification* (CAML) model [469], we refer to this method as LA_{CAML}.

An updated label-wise attention module was introduced in the *label attention model* (LAAT) [671]. We refer to this attention module as LA_{LAAT}. In LA_{LAAT}, the label-specific attention is computed similarly to LA_{CAML} as $A = \text{softmax}(UZ)$, where $U \in \mathbb{R}^{L \times d_p}$ is a learnable embedding matrix, but with $Z = \tanh(PH)$ where $P \in \mathbb{R}^{d_p \times d_h}$ is a learnable matrix, $Z \in \mathbb{R}^{d_p \times n}$ and d_p is a hyperparameter.

8.2.3 TRAINING AND EVALUATION METHODS

Mullenbach et al. [469] released code for pre-processing the discharge summaries, generating the train-test split, and evaluating model performance on MIMIC-III which many subsequent papers have used [30, 281, 338, 398, 671, 729]. Pre-processing consisted of lower-casing all text and removing words that only contain out-of-alphabet characters. Predicting procedures and diagnosis codes were treated as a single task. The dataset was split into training, validation, and test sets using random sampling, ensuring that no patient occurred in both the training and test set. The (non-stratified) random sampling lead to 54% of the ICD codes in MIMIC-III *full* not being sampled in the test set. This complicates the interpretation of results since these codes only contribute true negatives or false positives. Models are evaluated using the micro and macro average of the area under the curve of the receiver operating characteristics (AUC-ROC), F1-score, and precision@k.

While most papers use the pre-processing, train-test split, and evaluation metrics described above, they differ in several aspects of training. This may lead to performance differences unrelated to modeling choices which are undesirable when seeking to compare models. For instance, due to varying memory constraints of different models, documents are usually truncated to some maximum length. In the literature, this maximum varies from 2,500 to 4,000 words [281, 469, 671]. Furthermore, not all papers tune the prediction decision boundary but simply set it to 0.5, hyperparameter search ranges and sampling methods vary between works, and learning rate schedulers are only used in LAAT and PLM-ICD[398, 469]. In LAAT, the learning rate was decreased by 90% when the F1-score had not increased for five epochs. PLM-ICD used a schedule with linear warm up followed by linear decay.

All models except for PLM-ICD use Word2Vec embeddings pre-trained on the MIMIC-III training set. PLM-ICD uses a BERT encoder pre-trained on PubMed to encode the text in chunks of 128 tokens, and these contextualized embeddings are fed to a LA_{LAAT} layer.

Finally, all models compute independent code probabilities using sigmoid activation functions and optimize the binary cross entropy loss function during training. table 8.2 presents the selected models.

8.3 METHODS

Table 8.2: An overview of the compared models.

Model	Encoder	Decoder	Param
Bi-GRU [469]	Word2Vec, Bi-GRU	Max-pool	9.9M
CNN [469]	Word2Vec, CNN	Max-pool	10.3M
CAML [469]	Word2Vec, CNN	LA_{CAML}	6.1M
MultiResCNN [398]	Word2Vec, ResNet	LA_{CAML}	11.9M
LAAT [671]	Word2Vec, Bi-LSTM	LA_{LAAT}	21.9M
PLM-ICD [281]	BERT	LA_{LAAT}	138.8M

8.3.1 SELECTION CRITERIA

In this study, we included both models trained from scratch and models with components pre-trained on external corpora. We excluded models that use multi-modal inputs, such as medical code descriptions [30, 75, 338, 469, 671], code synonyms [729], code hierarchies [75, 717], or associated Wikipedia articles [27], because they introduced additional complexity without providing evidence for significant performance improvements [469, 633, 671]. We excluded works without publically available source code as the experiment descriptions often lacked important implementation details.

8.3.2 EVALUATION METRICS

Similar to previous work, we evaluated models using AUC-ROC, F1-score, and precision@k. Additionally, we introduced exact match ratio (EMR), R-precision, and mean average precision (MAP). The EMR is the percentage of instances where all codes were predicted correctly. This allowed us to measure how many documents were predicted perfectly, which is important for *fully automated* medical coding. Where precision@k is computed based on the top-k codes (i.e., k is

fixed), R-precision considers a number of codes equal to the true number of relevant codes. Thus, R-precision is useful when the number of relevant codes varies considerably between documents, which is the case for the MIMIC datasets. Finally, in contrast to all other metrics, MAP considers the exact rank of all relevant codes in a document.

Previous works calculated the macro F1-score as the harmonic mean of the macro precision and macro recall [281, 398, 469, 671]. Opitz and Burst [503] analyze macro F1 formulas common in multi-class and multi-label classification. They demonstrate that the above formulation is suboptimal, as it rewards heavily biased classifiers in unbalanced datasets. Therefore, as recommended by the authors, we calculated the macro F1-score as the arithmetic mean of the F1-score for each class. As seen in table 8.1, 54% of codes in MIMIC-III *full* are missing in the test set. Previous works set the F1-score of all the missing codes in the test set to 0, resulting in a misleadingly low macro F1-score. Because 54% of the codes are missing, the maximum possible macro F1-score is 46%. We ignored all codes not in the test set for our reproduction, essentially trading bias for variance. For our revised comparison, we resolved the issue by instead sampling new splits that reduce missing codes to a negligible fraction (see section 8.3.3) and ignoring the few that were still missing.

8.3.3   DEFINITION OF SPLITS

We define three new splits: MIMIC-III *clean*, MIMIC-IV *ICD-9*, and *ICD-10*. As described in section 8.3.2, 54% of the codes in MIMIC-III *full* are absent from the test set, which introduces significant bias in the model evaluation metrics. Therefore, we created a new MIMIC-III split to ensure that most codes are present in both the training and test set. Specifically, we removed codes with fewer than ten occurrences, doubled the test set size, and sampled the documents using multi-label stratified sampling [589]. We ensured that no patient occurred in both the training and test set, preprocessed the text, and considered procedures and diagnosis codes as a single task as done by Mullenbach et al. [469]. We based our new split on the v1.4 version of the dataset and refer to it as MIMIC-III *clean*. Using the same method, we created two splits for MIMIC-IV v2.2: one containing all documents labeled with ICD-9 codes and one with ICD-10 codes.

8.3.4   REPRODUCIBILITY EXPERIMENTS

We ran reproducibility experiments with all models to evaluate whether the results in the original works could be reproduced and to validate our reimplementations. We ran these experiments on MIMIC-III *full*, and *50* as in the original works [281, 398, 469, 671]. We used the hyperparameters reported in each paper

Table 8.3: Hyperparameters, maximum document lengths, and decision boundary tuning strategies used in the original works compared to the optimal settings found in this paper (marked with *). LR is the learning rate scheduler. "Length" is the maximum number of words a document can contain before being truncated. † applies to models using word-piece tokenization. These models were filtered on the number of sub-words instead of full words. "DB tune" is whether the optimal decision boundary was found using the validation set. If a paper did not tune the decision boundary, it was set to 0.5.

Model	Batch Size	Weight Decay	Learning Rate	Dropout	LR Scheduler	Optimizer	Epochs	Length	DB tune
				Hyperparameters					
Bi-GRU	16	0.0	0.003	0.2	no	Adam	100	2500	no
Bi-GRU*	8	0.0001	0.001	0	yes	AdamW	20	4000	yes
CNN	16	0.0	0.003	0.2	no	Adam	100	2500	no
CNN*	8	0.00001	0.001	0	yes	AdamW	20	4000	yes
CAML	16	0.0	0.0001	0.2	no	Adam	200	2500	no
CAML*	8	0.001	0.005	0.2	yes	AdamW	20	4000	yes
MultiResCNN	16	0.0	0.0001	0.2	no	Adam	200	2500	no
MultiResCNN*	16	0.0001	0.0005	0.2	yes	AdamW	20	4000	yes
LAAT	8	0.0	0.0001	0.3	yes	AdamW	50	4000	no
LAAT*	8	0.001	0.001	0.2	yes	AdamW	20	4000	yes
PLM-ICD	8	0.0	0.00005	0.2	yes	AdamW	20	3072†	yes
PLM-ICD*	16	0.0	0.00005	0.2	yes	AdamW	20	4000	yes

(see table 8.3) and report both the original and the revised macro F1-scores discussed in section 8.3.2.

8.3.5 REVISED COMPARISON

To address the issues associated with comparing results reported by previous works described in sections 8.2.3 and 8.3.2, we perform a revised model comparison. We run experiments on the new MIMIC-III *clean*, MIMIC-IV *ICD-9*, and *ICD-10*. All models were trained for 20 epochs using a learning rate schedule with linear warm up for the first 2K updates followed by linear decay [281]. We found this schedule to speed up the training convergence of all the models. Whereas original works use Adam or AdamW, we used AdamW for all experiments as it corrects the weight decay implementation of Adam [341, 426]. For each model, we tuned the decision boundary to maximize the micro F1-score on the validation set. We used randomized sampling to find optimal settings for dropout, weight decay, learning rate, and batch size. The hyperparameter search was performed on MIMIC-III *clean*, and the MIMIC-IV splits. We found that the best setting for each model generalized across datasets. Using this setting, we ran each model ten times with different seeds on each dataset. All documents were truncated to a maximum of 4000 words. The hyperparameters, maximum document lengths, and decision boundary tuning strategy are summarized in table 8.3.

We performed an ablation study to analyze which changes had the largest impact on performance. Specifically, we evaluated the effect of truncation, hyperparameter search, and decision boundary tuning. We modified one of these at a time: We ran one experiment where documents were truncated to a maximum length of 2,500 words, a second experiment where the models were trained with the hyperparameters, number of epochs, and learning rate schedule used in the original works, and a third experiment where the decision boundary was set to 0.5 instead of tuned.

8.3.6 ERROR ANALYSIS

To validate and falsify the commonly cited challenges of AMC, which include a lack of training data, long documents, and rare codes, we performed an error analysis. In addition to analyzing rare codes, we contribute an in-depth code analysis aiming to identify the attributes that make certain codes challenging to predict.

Amount of training data: Multiple studies attribute poor performance to data sparsity of MIMIC-III, which contains only fifty thousand examples [330, 634,

719, 724]. MIMIC-IV *ICD-9* contains four times as many examples, which allows analyzing the effect of training set size. We train each model on 25k, 50k, 75k, 100k, and 125k examples and report micro and macro F1 on the fixed test set. The training subsets were sampled from the training set using multi-label stratified sampling to ensure the same code distributions [589].

Document length: We analyzed whether model performance correlates with document length on MIMIC-IV *ICD-9*. Specifically, we calculated the Pearson and Spearman correlation between the number of words in the documents and the micro F1-score for all models. For each model, we used the best seed from the revised comparison.

Code analysis: To analyze the performance impact of rare codes, we first calculated the Pearson and Spearman correlation between model performance on each code and the corresponding code frequency in the training data. We calculated these correlations for all splits. To identify attributes of challenging codes, we analyzed model performance on the chapter level of the ICD-10 classification system. Using high-level chapters instead of codes allows us to group examples into categories, which we use as a starting point for further analysis. We limit the scope of the analysis to diagnosis codes. We focused on ICD-10 because it is the classification system currently in use at most hospitals.

8.4   Results

8.4.1   Reproduced results

In table 8.4, we report the reproduced results on MIMIC-III *full* and *50* using hyperparameters as reported in the original papers. We list the original and corrected macro F1-score described in section 8.3.2. In most cases, our corrections doubled the macro F1-scores on MIMIC-III *full*. The differences were smaller on MIMIC-III *50* because all included codes are in the test set.

8.4.2   Revised comparison

The results of our revised comparison on MIMIC-III *clean*, MIMIC-IV *ICD-9*, and *ICD-10* are shown in table 8.5. Contrary to the originally reported results, Bi-GRU performs better than CNN in all metrics. Otherwise, the model performance ranking is unchanged from the original works. PLM-ICD outperformed the other models on all metrics and all datasets. The models previously reported as least performant improved the most.

Table 8.4: Reproduced test set results compared with those from the original works. Our reproduced results are indicated with *. The results were reproduced on MIMIC-III v1.4 with the preprocessing pipeline and splits of Mullenbach et al. [469]. Each model was reproduced using the hyperparameters presented in the respective paper. We use both macro F1 formulas: Macro† refers to the method used in the original work, while Macro refers to the corrected version used in this paper.

| | MIMIC-III _full_ | | | | | | | MIMIC-III 50 | | | | | |
| | AUC-ROC | | F1 | | | Precision@k | | AUC-ROC | | F1 | | | Precision@k |
	Micro	Macro	Micro	Macro†	Macro	8	15	Micro	Macro	Micro	Macro†	Macro	5
CNN	96.9	80.6	41.9	4.2	-	58.1	48.8	90.7	87.6	62.5	57.6	-	62.0
CNN*	97.3	83.1	41.5	3.4	6.7	61.9	47.2	91.9	89.2	64.9	58.8	58.0	62.6
Bi-GRU	97.1	82.2	41.7	3.8	-	58.5	44.5	89.2	82.8	54.9	48.4	-	59.1
Bi-GRU*	98.0	87.1	42.6	3.6	7.0	65.0	49.8	89.3	85.2	56.1	46.2	43.1	57.9
CAML	98.6	89.5	53.9	8.8	-	70.9	56.1	90.9	87.5	61.4	53.2	-	60.9
CAML*	98.4	88.4	49.5	5.6	11.3	69.9	54.9	91.1	87.5	60.6	52.4	51.0	61.1
MultiResCNN	98.6	91.0	55.2	8.6	-	73.4	58.4	93.8	89.9	67.0	60.6	-	64.1
MultiResCNN*	98.6	90.8	56.5	9.2	18.5	73.4	58.4	92.4	89.7	67.3	62.2	61.1	63.4
LAAT	98.8	91.9	57.5	9.9	-	74.5	59.1	94.6	92.5	71.5	66.6	-	67.5
LAAT*	98.6	89.5	56.1	8.2	16.2	73.9	58.7	92.8	90.5	66.8	60.8	59.2	64.0
PLM-ICD	98.9	92.6	59.8	10.4	-	77.1	61.3	-	-	-	-	-	-
PLM-ICD*	98.8	92.3	58.9	11.1	22.8	75.7	60.5	93.8	91.7	70.5	66.3	65.4	65.7

Table 8.5: Results on the MIMIC-III *clean*, MIMIC-IV *ICD-9* and MIMIC-IV *ICD-10* test sets presented as percentages. Micro F1-scores rank the table in ascending order. Each model was trained ten times with different seeds. We performed a McNemar's test with Bonferroni correction and found that all the models are significantly different ($p < 0.001$).

| | | Classification | | | | | Ranking | | | |
| | | AUC-ROC | | F1 | | EMR | Precision@k | | R-precision | MAP |
		Micro	Macro	Micro	Macro		8	15		
MIMIC-III clean	CNN	97.1±0.0	88.1±0.2	48.0±0.3	9.9±0.4	0.1±0.0	61.6±0.2	46.6±0.1	49.1±0.2	50.6±0.2
	Bi-GRU	97.8±0.1	91.1±0.2	49.7±0.4	12.2±0.2	0.1±0.0	62.8±0.4	47.6±0.4	50.1±0.4	52.1±0.4
	CAML	98.2±0.0	91.4±0.2	55.4±0.1	20.4±0.3	0.1±0.0	67.7±0.2	52.8±0.1	55.8±0.1	58.9±0.2
	MultiResCNN	98.5±0.0	93.1±0.3	56.4±0.2	22.9±0.6	0.1±0.0	68.5±0.2	53.5±0.1	56.7±0.2	59.9±0.3
	LAAT	98.6±0.1	94.0±0.3	57.8±0.2	22.6±0.6	0.2±0.1	70.1±0.2	54.8±0.2	58.0±0.2	61.7±0.3
	PLM-ICD	**98.9±0.0**	**95.9±0.1**	**59.6±0.2**	**26.6±0.8**	**0.4±0.0**	**72.1±0.2**	**56.5±0.1**	**60.1±0.1**	**64.6±0.2**
MIMIC-IV ICD-9	CNN	98.1±0.1	89.4±0.5	52.4±0.1	12.6±0.4	0.6±0.0	61.3±0.1	45.6±0.0	52.9±0.1	55.2±0.1
	Bi-GRU	98.8±0.0	93.8±0.1	55.5±0.1	16.6±0.2	0.7±0.0	64.1±0.1	47.8±0.1	55.8±0.1	58.9±0.1
	CAML	98.8±0.0	90.7±0.3	58.6±0.1	19.3±0.1	0.6±0.0	66.3±0.1	50.3±0.0	58.5±0.1	62.4±0.1
	MultiResCNN	99.2±0.0	95.1±0.1	60.4±0.0	27.7±0.3	0.8±0.0	67.6±0.0	51.8±0.0	60.4±0.0	64.7±0.1
	LAAT	99.3±0.0	96.0±0.3	61.7±0.1	26.4±0.9	0.9±0.0	68.9±0.1	52.7±0.1	61.7±0.2	66.3±0.2
	PLM-ICD	**99.4±0.0**	**97.2±0.2**	**62.6±0.3**	**29.8±1.0**	**1.0±0.1**	**70.0±0.2**	**53.5±0.2**	**62.7±0.3**	**68.0±0.3**
MIMIC-IV ICD-10	CNN	97.5±0.1	87.9±0.4	47.2±0.6	8.0±0.4	0.3±0.0	60.3±0.1	45.7±0.1	47.3±0.2	48.2±0.2
	Bi-GRU	98.3±0.0	92.4±0.2	50.1±0.2	10.6±0.4	0.3±0.0	62.6±0.2	47.7±0.2	49.6±0.1	51.1±0.2
	CAML	98.5±0.0	91.1±0.1	55.4±0.2	16.0±0.3	0.3±0.0	66.8±0.2	52.2±0.1	54.5±0.2	57.4±0.2
	MultiResCNN	99.0±0.0	94.5±0.2	56.9±0.1	**21.1±0.2**	**0.4±0.0**	67.8±0.1	53.5±0.1	56.1±0.1	59.3±0.2
	LAAT	99.0±0.1	95.4±0.3	57.9±0.1	20.3±0.4	**0.4±0.0**	68.9±0.1	54.3±0.1	57.2±0.1	60.6±0.2
	PLM-ICD	**99.2±0.0**	**96.6±0.2**	**58.5±0.7**	21.1±2.3	**0.4±0.0**	**69.9±0.6**	**55.0±0.6**	**57.9±0.8**	**61.9±0.9**

The ablation study results are shown in table 8.6 for MIMIC-III *clean*. Truncating the documents to 2,500 words instead of 4,000 had little impact on the performance. Using the hyperparameters from the original works degraded the performance substantially for CAML, Bi-GRU, and CNN but had a smaller effect on the other models. Not tuning the decision boundary had the largest negative effect on all models except MultiResCNN. In figure 8.2, we plot the relationship between the decision boundary and F1-scores. LAAT and MultiResCNN perform similarly when using a decision boundary of 0.5. However, when tuning the decision boundary, LAAT outperforms MultiResCNN considerably. Similar results were obtained on the other datasets.

8.4.3 ERROR ANALYSIS

Amount of training data: figure 8.3 shows the relationship between the number of training examples and the micro and macro F1-scores for all models. In most cases, increasing the training data had a larger effect on the macro F1-score than the micro F1-score, indicating more extensive improvements in rare codes than common codes. The curve for macro F1 is less smooth because the decision boundary was tuned on the micro F1-scores.

Document length: We plot the micro F1-score for all models as a function of the number of words per document in figure 8.4. We note that all models underperformed on documents with fewer than 1000 words. By manual inspection, we found that most of these documents missed the information necessary to predict their labeled codes, leading to underperformance. In table 8.7, we list the Pearson and Spearman correlations. We excluded documents shorter than 1000 words to avoid confounding with missing information and longer than 4000 words due to the truncation limit. We observe a very small negative correlation between document length and micro F1 which matches the downward trend in micro F1, starting from approximately 1000 words in figure 8.4. Although document length may itself be the cause of the slightly lower performance for long documents, there may be other factors correlated with document length impacting performance, such as the number of codes per document and code frequency. As there are few long documents, the effect on average micro F1 for each dataset is negligible; hence, previous claims that long documents lead to poor performance in AMC could not be validated. Results on MIMIC-IV *ICD-10* and MIMIC-III *clean* were similar.

Code analysis: Figure 8.5 compares the best performing model, PLM-ICD, trained and evaluated on MIMIC-IV *ICD-9* and *ICD-10*. Similar results were obtained on MIMIC-III *clean*. The comparison shows the relationship between code frequen-

Table 8.6: Ablation study on MIMIC-III *clean*. The numbers are the micro/macro F1-scores on the test set.

	PLM-ICD	LAAT	MultiResCNN	CAML	Bi-GRU	CNN
Our result	59.6/26.6	57.8/22.6	56.4/22.9	55.4/20.4	49.7/12.2	48.0/9.9
Input length truncated at 2500 words	59.4/26.2	57.6/22.3	56.0/23.2	54.8/19.7	49.4/12.0	47.9/9.8
No decision boundary tuning	58.7/23.0	56.2/19.0	56.2/22.6	53.3/17.1	45.3/8.1	43.8/7.0
Original hyperparameters	59.6/27.0	57.5/21.6	56.4/20.0	52.8/17.3	48.1/11.2	46.9/10.2
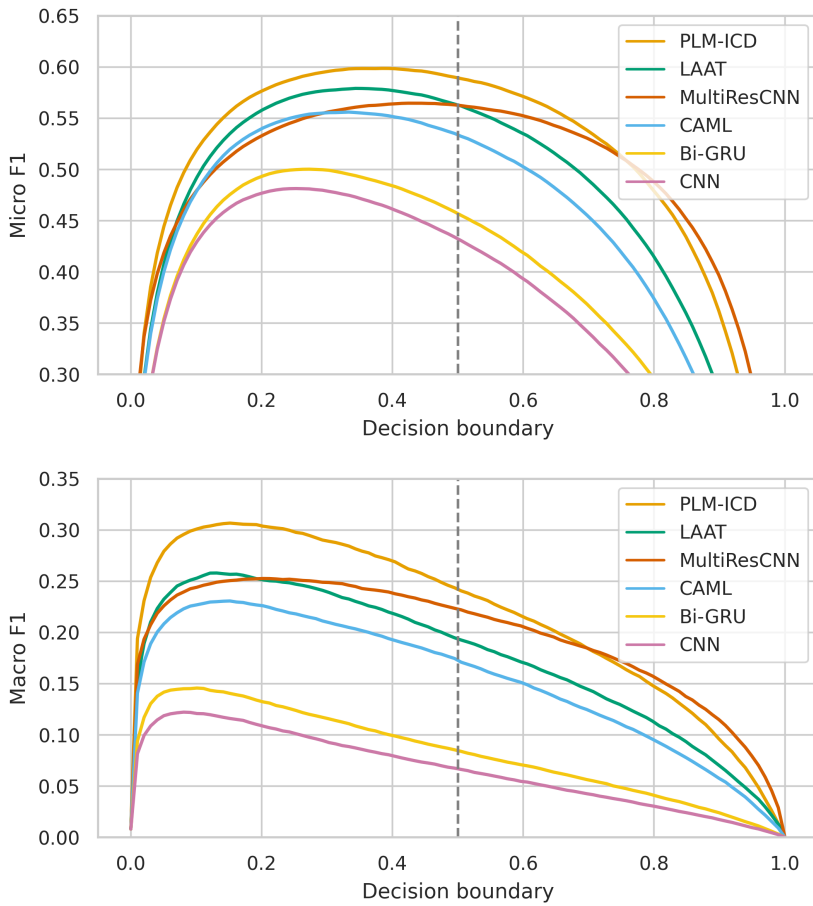
Figure 8.2: The relationship between chosen threshold and F1-score of every reproduced model in table 8.4. The left figure shows the micro F1-score, and the right shows the macro F1-score. The models were evaluated on MIMIC-III *clean*.
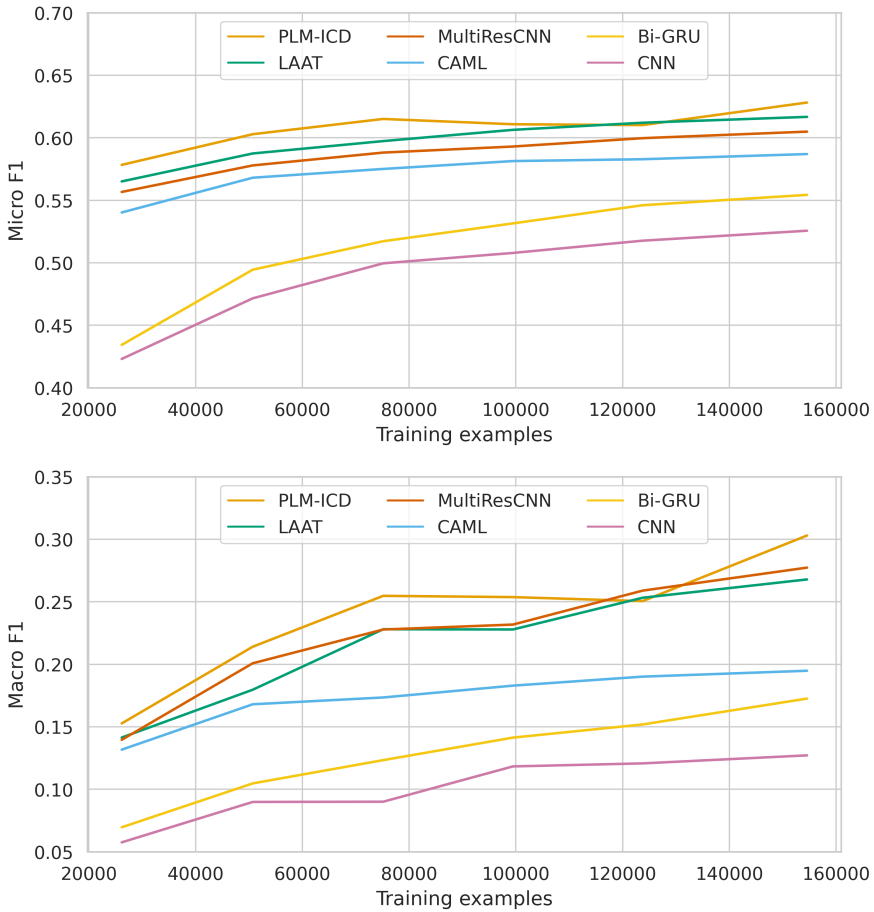
Figure 8.3: The relationship between the number of training examples and F1-score on MIMIC-IV *ICD-9*. The left figure shows the F1 Micro score on the y-axis, while the right figure shows the F1 Macro score.
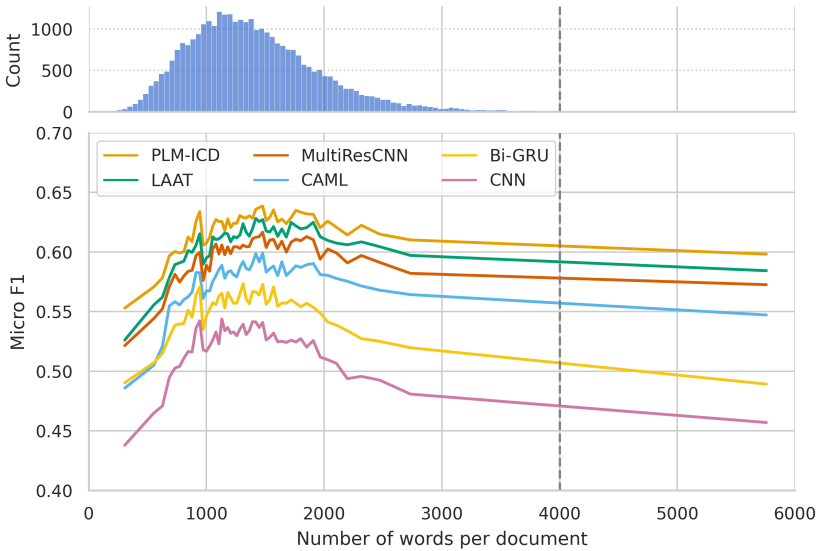
Figure 8.4: Relationship between the lengths of the clinical notes and the micro F1-score for each model on MIMIC-IV *ICD-9*. The vertical line indicates the maximum length of the notes after truncation. The histogram at the top visualizes the document length distribution.

Table 8.7: Correlation between the F1-score and the logarithm of code frequency and document length on MIMIC-IV *ICD-9*. As discussed in section 8.4.3, we only considered document lengths between 1000 and 4000 words. All correlations are statistically significant (p < 0.001).

	Code frequency		Document lengths	
	Pearson	Spearman	Pearson	Spearman
CNN	0.61	0.68	-0.09	-0.08
Bi-GRU	0.57	0.65	-0.08	-0.07
CAML	0.56	0.60	-0.03	-0.03
MultiResCNN	0.47	0.53	-0.02	-0.03
LAAT	0.52	0.57	-0.02	-0.02
PLM-ICD	0.48	0.52	-0.02	-0.02

cies in the training set and macro F1-scores. As shown in table 8.5, all models perform worse on *ICD-10* compared to *ICD-9*. However, figure 8.5 demonstrates that performance on codes with similar frequencies is comparable between the two splits. This suggests that the performance differences in table 8.5 are due to *ICD-10* containing a higher fraction of rare codes as shown in figures 8.1 and 8.5.

The Pearson and Spearman correlations between the logarithm of code frequency and F1-score are shown in table 8.7 for MIMIC-IV *ICD-9*. Similar correlations were observed for the other datasets. All the models show moderately high correlation confirming that performance on rare codes is generally lower than on common codes. To further our understanding of the problem, we computed the percentage of unique codes in each dataset that the models never predicted. As seen in table 8.8, no model correctly predicted more than 50% of the ICD-10 codes.

Figure 8.6 shows the performance of PLM-ICD on each ICD-10 chapter—the top-most level in the tree-like hierarchy. For our analysis, we limited the scope to only focus on the diagnosis codes. We also excluded codes with fewer than one hundred training examples to control for some chapters having many rare codes.

Overall, PLM-ICD never correctly predicted 2,928 of the 5,794 ICD-10 diagnosis codes in our split. Of these codes, only 110 had over a hundred training examples, and 58 belong to only two of the 20 chapters in MIMIC-IV *ICD-10*. Specifically, 45 belong to the chapter relating to "external causes of morbidity"(Z00-Z99), while 13 relate to "factors influencing health status and contact with health services"(V00-Y99). To further investigate why most non-predicted codes with more than 100 training examples belong to only two chapters, we manually inspected a selection of codes in these chapters, as described in the following.

The Z68 category, part of the Z00-Z99 chapter, contains codes related to the patient's body mass index (BMI). Codes within this category occur more than 17,000 times in the MIMIC-IV training data, but PLM-ICD never predicts 20 out of the 26 codes of Z68. One possible hypothesis is that PLM-ICD struggles with extracting the BMI from the discharge summaries, as all digits have been removed in the pre-processing. We found several other codes containing digits in the code descriptions that the model failed to detect, e.g., "Blood alcohol level of less than 20 mg/100 ml"(Y90.0), "34 weeks gestation of pregnancy"(Z3A.34), and "NIHSS score 15"(R29.715). These observations support our hypothesis that removing digits in the pre-processing makes certain codes challenging to predict.

The Y92 category, part of the V00-Y99 chapter, contains codes related to the physical location of occurrence of the external cause. It is a large category of 246 unique codes occurring 27,870 times in the training set. The category is challenging due to locations being very specific. For instance, there are unique codes for whether an incident occurred on a tennis court, squash court, art gallery, or mu-

seum. We hypothesize that the level of detail in the discharge summaries does not always match the fine-grained code differences.
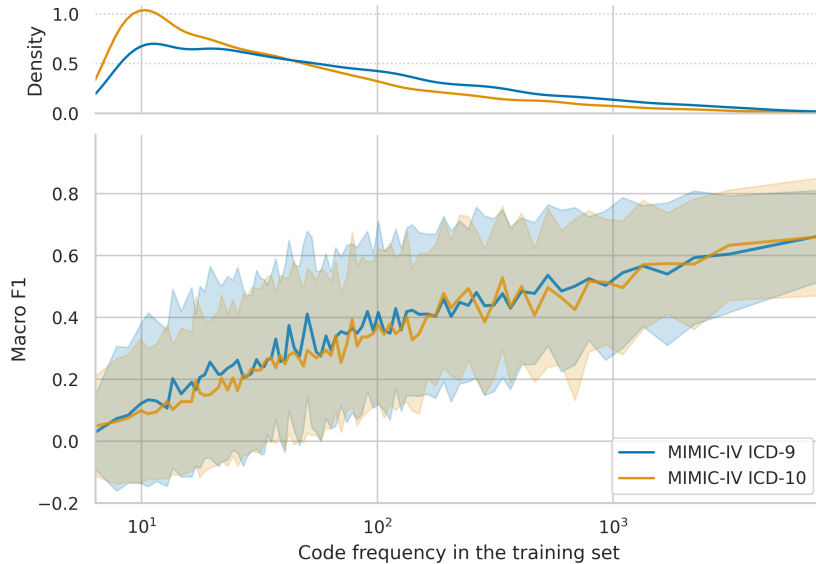
Figure 8.5: Relationship between the code frequencies in the training set and the macro F1-score for PLM-ICD on MIMIC-IV *ICD-9* and *ICD-10*. The shaded area indicates the standard deviation of the score computed for codes within the bin.

There are ten different codes in MIMIC-IV *ICD-10* relating to nicotine dependence and tobacco use. The three most common are Z87.891 ("Personal history of nicotine dependence"), F17.210 ("Nicotine dependence, cigarettes, uncomplicated"), and Z72.0 ("Tobacco use"), with 26,427, 8,486, and 1,914 training examples, respectively. Among these, Z72.0 was the third most common single code in the training set that PLM-ICD never predicted correctly. PLM-ICD achieved an F1-score of 53% for Z87.891, 51% for F17.210, and 0% for Z72.0 and all other nicotine-related codes. These findings suggest that when there is a class imbalance among highly similar codes, PLM-ICD is strongly biased toward the most frequent ones.

Table 8.8: Percentage of ICD diagnosis codes in the test set that the models never predicted correctly.

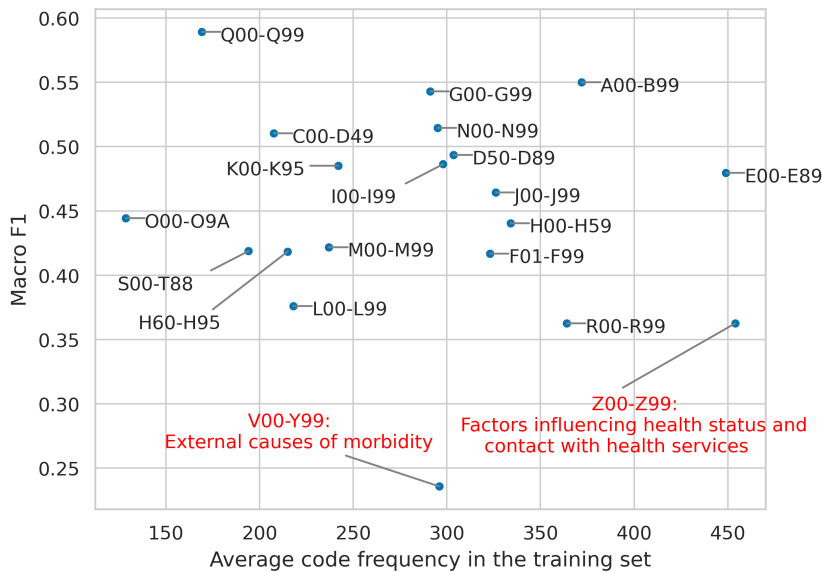|	MIMIC-III	MIMIC-IV	
	clean	*ICD-9*	*ICD-10*
CNN	68.2	61.5	72.0
Bi-GRU	65.0	54.3	67.1
CAML	52.8	57.0	62.0
MultiResCNN	48.8	40.3	53.5
LAAT	50.4	43.6	55.0
PLM-ICD	44.3	39.3	51.8

Figure 8.6: Performance of PLM-ICD on ICD-10 chapters. Only codes with more than a hundred occurrences in the MIIMC-IV ICD-10 training set were considered, leaving 20 chapters. We found Z00-Z99 and V00-Y99 to be the most challenging.

8.5   DISCUSSION

8.5.1   LESSONS LEARNED

We found reproducing the results of CNN, Bi-GRU, CAML, and LAAT challenging. While we expected discrepancies due to random weight initializations and data shuffling, the differences from the original works exceeded our presuppositions. Our reproduced results were better than originally reported for Bi-GRU and CNN and worse for CAML and LAAT on most metrics. There have been multiple reports of issues in reproducing the results of Mullenbach et al. [469].[9] Additionally, most previous works did not report which version of MIMIC-III they used, and the code and hyperparameter configurations were not documented in detail. Therefore, we hypothesize that our results differ because previous works report incorrect hyperparameters or use an earlier version of MIMIC-III.

We showed that models previously reported as low-performing underperformed partly due to a poor selection of hyperparameters and not tuning the decision boundary. In our revised comparison, we demonstrated that training the models using our setup decreased the difference between the best and worst micro F1-scores by 5.8 percentage points. Mullenbach et al. [469] concluded that CNN outperformed Bi-GRU. However, in our revised comparison, Bi-GRU outperformed CNN on all metrics on MIMIC-III *clean*, MIMIC-IV *ICD-9*, and MIMIC-IV *ICD-10*.

Even though MultiResCNN contains more parameters than CAML, Li and Yu [398] concluded that MultiResCNN was faster to train because it converged in fewer epochs. However, this was only true when using the original setup where CAML converged after 84 epochs. We found that when using a learning rate schedule and appropriate hyperparameters, it was possible to train all the models in 20 epochs without sacrificing performance. Therefore, with our setup, CAML was faster to train than MultiResCNN.

We demonstrated that the macro F1-score had been underestimated in prior works due to the poorly sampled MIMIC-III *full* split and the practice of setting the F1-score of all codes absent in the test set to 0. Since 54% of the codes in MIMIC-III *full* are missing in the test set, the maximum possible macro F1-score is 46%. The previously highest reported macro F1-score on MIMIC-III *full* is 12.7% for PLM-ICD [338]. Using our corrected macro F1-score on the same split, PLM-ICD achieved a macro F1-score of 22.8%. This large difference from previous state-of-the-art seems to indicate that all previous work on AMC used the suboptimally calculated macro F1-score, including works not reproduced in this paper. Many studies use the macro F1-score to evaluate the ability of their models to predict rare codes [338, 729]. If it has indeed been incorrectly calculated

---

[9]https://github.com/jamesmullenbach/caml-mimic

in these studies, some conclusions drawn in previous work regarding rare code prediction may have been misguided.

Multiple studies mention lack of training data, rare codes, and long documents as the main challenges of AMC [160, 181, 281, 307, 398, 419, 464, 515, 633, 634, 666, 671]. In the error analysis, we aimed to validate or falsify these assumptions. We found that rare codes were challenging for all models and observed that more than half of all ICD-10 codes were never predicted correctly. Furthermore, in figure 8.3, we showed that when adding more training data, most models see a greater performance improvement on rare codes than on common codes. These findings suggest that medical coding is fundamentally challenged by a lack of training data that, in turn, gives rise to many rare codes. We found that document length and model performance only exhibited a weak correlation. Specifically, the low number of very long documents was insufficient to affect the average performance on the dataset.

8.5.2 Future work

We recommend future work within AMC use our revised comparison method, including stratified sampled splits of MIMIC datasets, corrected evaluation metrics, hyperparameter search, and decision boundary tuning to avoid reporting suboptimal or biased results. Furthermore, for AMC to become a viable solution for ICD-10, future research should focus on improving performance on rare codes while, in the shorter term, developing methods to detect codes that are too challenging for automated coding and, therefore, should be coded manually. Finally, while PLM-ICD outperforms the other models in this paper, the improvements are limited compared to the effect of pre-training in other domains [26, 151, 161, 404, 461]. Notably, there have been several unsuccessful attempts at using pre-trained transformers for medical coding [194, 307, 454, 515, 740]. In future work, we want to investigate why pre-trained transformers underperform in medical coding.

8.5.3 Limitations

We presented findings and analyses on MIMIC-III and MIMIC-IV. It is unclear how our findings generalize to medical coding in real-world settings. For instance, since MIMIC-III and IV contain data from the emergency department and ICU of a single hospital, the findings in this paper may not generalize to other departments or hospitals. For instance, discharge summaries from outpatient care are often easier to code than summaries from inpatient care as they are shorter with fewer codes per document [419, 650, 740].

The medical code labeling of MIMIC is used as a gold standard in this paper. However, medical coding is error-prone, and, in many cases, deciding between certain codes can be a subjective matter [425, 491]. Burns et al. [72] systematically reviewed studies assessing the accuracy of human medical coders and found an overall median accuracy of 83.2% (IQR: 67.3-92.1%). Searle, Ibrahim, and Dobson [588] investigated the quality of the human annotations in MIMIC-III and concluded that 35% of the common codes were under-coded. Such errors and subjectivity in manual medical coding make model training and evaluation challenging and suggests that additional evaluation methods using, e.g., a human-in-the-loop, could be useful to increase the reliability of results.

8.6 CONCLUSION

In this paper, we first reproduced the results of selected state-of-the-art models focusing on unimodal models with publically available source code. We found that model evaluation in original works was biased by an inappropriate formulation of the macro F1-score and treatment of missing classes in the test set. By fixing the macro F1 computation, we approximately doubled the macro F1 of the reproduced models on MIMIC-III *full*. We introduced a new *clean* split for MIMIC-III that contains all classes in the test set and performed a revised comparison of all models under the same training, evaluation, and experimental setup, including hyperparameter and decision boundary tuning. We observed a significant performance improvement for all models, with those previously reported as low-performing improving the most. We reported the first results of current state-of-the-art models on the newly released MIMIC-IV dataset [209, 314] and provided splits for the *ICD-9* and *ICD-10* coded subsets using the same method as for MIMIC-III *clean*. Through error analysis, we provided empirical evidence for multiple model weaknesses. Specifically, models underperform severely on rare codes and, in contrast to previous claims, long documents only have a negligible negative performance impact. We release our source code, model parameters, and the new MIMIC-III *clean* and MIMIC-IV *ICD-9* and *ICD-10* splits.[(7)]

A Retrospective Study on Machine Learning-Assisted Stroke Recognition for Medical Helpline Calls

This chapter is a piece of original research by the candidate, previously published elsewhere and reprinted here with permission:

Abstract

Advanced stroke treatment is time dependent and, therefore, relies on recognition by call-takers at prehospital telehealth services to ensure fast hospitalization. This study aims to develop and assess the potential of machine learning in improving prehospital stroke recognition during medical helpline calls. We use calls from 1 January 2015 to 31 December 2020 in Copenhagen to develop a machine learning-based classification pipeline. Calls from 2021 are used for testing. Calls are first transcribed using an automatic speech recognition model and then categorized as stroke or non-stroke using a text classification model. Call-takers achieve a sensitivity of 52.7% (95% confidence interval 49.2-56.4%) with a positive predictive value (PPV) of 17.1% (15.5-18.6%). The machine learning framework performs significantly better (p < 0.0001) with a sensitivity of 63.0% (62.0-64.1%) and a PPV of 24.9% (24.3-25.5%). Thus, a machine learning framework for recognizing stroke in prehospital medical helpline calls may become a supportive tool for call-takers, aiding in early and accurate stroke recognition.

9.1 Introduction

Stroke is a leading cause of disability and death worldwide [199, 329, 360]. Effective treatment is time-sensitive, and an optimal outcome is more likely when treatment is administered within the first four and a half hours from stroke onset [46, 652]. The gateway to ambulance transport and hospital admittance is

through prehospital telehealth services, including emergency medical call centers, nurse advice call lines, and out-of-hours health services. In the pre-hospital setting, the use of mobile stroke units has made it possible to deliver advanced treatment faster [236, 476]. As the mobile stroke unit is only dispatched to patients with a suspected stroke, the impact of mobile stroke unit is directly influenced by accurate call-taker recognition of stroke [236, 476]. Call-takers who can rapidly and accurately recognize stroke are therefore crucial in facilitating prompt care in both pre-hospital and in-hospital settings.

Despite initiatives to improve stroke recognition [213, 353], approximately half of all patients with stroke do not receive the correct triage for their condition from call-takers [54, 502, 668]. Most initiatives aim to improve stroke recognition by call-takers via introducing more specific assessment tools [213, 353] or providing specialized training [696]. Recent advances in machine learning technology might be applied to improve stroke recognition without requiring changes to the triaging approach, and machine learning aided identification of stroke has been suggested as a means of improving mobile stroke unit effectiveness [476]. Real-time feedback from a machine learning model can improve the recognition of out-of-hospital cardiac arrest [52, 53]. Therefore, this study aimed to develop and assess the potential of machine learning in improving prehospital stroke recognition during medical helpline calls.

In this study, we use call recordings and registry data from the Copenhagen Emergency Medical Services (CEMS) and the Danish Stroke Registry (DanStroke) [313] from 2015 to 2020. We obtain call recordings from two call lines: the 1-1-2 emergency line and the medical helpline 1813 (MH-1813). We then fit a machine learning framework to classify medical helpline calls as stroke or non-stroke. Calls are first transcribed using an automatic speech recognition model and then categorized by a text classification model trained as an ensemble of five individual models. We compare the performance of the model with that of call-takers using MH-1813 data from 2021.

9.2 RESULTS

9.2.1 POPULATION CHARACTERISTICS

Calls to the MH-1813 were divided into training, validation, and test subsets and calls to the emergency line 1-1-2 were only used as supplementary training data (table 9.1). Calls from the test year (2021) that were not associated with a diagnostic category code, which we used to evaluate call-taker performance, were separated from our primary test set, but still included to assess potential bias in this group of calls (2021 w/o category, table 9.1). The 1-1-2 training data differed from the MH-1813 data regarding age, male/female ratio, and stroke prevalence

Table 9.1: Population characteristics for each data subset.

	Training (112)	Training (MH-1813)	Validation	Test	2021 w/o category
All calls					
Num. calls	155,696	1,391,301	155,825	344,030	231,009
Female	74,640 (47.94%)	792,783 (56.98%)	86,959 (55.81%)	190,974 (55.51%)	134,324 (58.14%)
Male	79,564 (51.10%)	596,760 (42.89%)	68,866 (44.19%)	153,050 (44.49%)	96,258 (41.67%)
65+ years	72,930 (46.84%)	335,146 (24.09%)	30,313 (19.45%)	65,652 (19.08%)	81,488 (35.27%)
Age (mean ± std.)	59.47 ± 21.24	47.12 ± 21.38	44.63 ± 20.08	44.31 ± 20.10	50.36 ± 22.77
Stroke calls					
Num. calls	3,899	3,471	360	757	679
Female	1,784 (45.76%)	1,654 (47.65%)	161 (44.72%)	349 (46.10%)	366 (53.90%)
Male	2,115 (54.24%)	1,815 (52.29%)	199 (55.28%)	408 (53.90%)	313 (46.10%)
65+ years	2,968 (76.12%)	2,421 (69.75%)	250 (69.44%)	555 (73.32%)	567 (83.51%)
Age (mean ± std.)	72.91 ± 12.77	70.68 ± 13.85	70.93 ± 13.83	71.51 ± 13.41	73.41 ± 14.11
Non-stroke calls					
Num. calls	151,797	1,387,830	155,465	343,273	230,330
Female	72,856 (48.00%)	791,129 (57.00%)	86,798 (55.83%)	190,625 (55.53%)	133,958 (58.16%)
Male	77,449 (51.02%)	594,945 (42.87%)	68,667 (44.17%)	152,642 (44.47%)	95,945 (41.66%)
65+ years	69,962 (46.09%)	332,725 (23.97%)	30,063 (19.34%)	65,097 (18.96%)	80,921 (35.13%)
Age (mean ± std.)	59.12 ± 21.30	47.06 ± 21.36	44.57 ± 20.05	44.25 ± 20.08	50.29 ± 22.76

(table 9.1). We therefore performed an ablation study where 1-1-2 data were not used for training to assess whether this difference negatively impacted model performance. The training, validation, and test subsets of the MH-1813 data had similar characteristics, whereas the 2021 data without diagnostic categories differed in age and sex.

9.2.2 Main results

The classification model outperformed the call-takers (table 9.2), with significant differences in all metrics ($p < 0.0001$, paired approximate permutation test). Excluding the 1-1-2 call line training data significantly degraded the model's performance ($p < 0.0001$, paired approximate permutation test), despite the domain mismatch with the MH-1813 call line test data. The performance on the 2021 calls without a diagnostic category was significantly worse than that of the test set regarding F1-score, sensitivity, false positive rate (FPR), and false omission rate (FOR) ($p < 0.0001$, independent approximate permutation test). The difference in positive predictive value (PPV) was not significant ($p = 0.298$, independent approximate permutation test).

The receiver operating characteristic (ROC) curve (figure 9.1, left) illustrates the potential to increase the sensitivity while maintaining a FPR lower than or equal to that of the call-takers. Similarly, the PPV-sensitivity curve (figure 9.1, right) demonstrates that sensitivity can be improved while retaining a PPV higher than that of the call-takers. The framework can thus be tuned to a sensitivity of around 73%, while still having a higher positive predictive value than the human

Table 9.2: Overall performance on MH-1813 test data, performance without 1-1-2 training data, and performance on data from 2021 without diagnostic categories as well as performance on MH-1813 based on demographic subgroups (age/sex) [mean (95% CI)]. NPV: negative predictive value, PPV: positive predictive value, FOR: false omission rate, CI: confidence interval.

	F1-score [%] ↑	Sensitivity [%] ↑	PPV [%] ↑	FOR [%] ↓ (1 - specificity)	FPR [%] ↓ (1 - NPV)
Overall					
Call-takers	25.8 (23.7-27.9)	52.7 (49.2-56.4)	17.1 (15.5-18.6)	0.105 (0.094-0.116)	0.565 (0.539-0.590)
Model	35.7 (35.0-36.4)	63.0 (62.0-64.1)	24.9 (24.3-25.5)	0.082 (0.079-0.085)	0.419 (0.413-0.426)
Without 112 training data					
Model	32.4 (31.8-33.1)	60.4 (59.3-61.4)	22.2 (21.6-22.7)	0.088 (0.085-0.091)	0.467 (0.460-0.474)
On MH-1813 data without diagnostic category					
Model	32.6 (31.9-33.4)	48.3 (47.2-49.4)	24.7 (23.9-25.3)	0.153 (0.148-0.158)	0.435 (0.427-0.443)
18-64 years					
Call-takers	15.9 (13.1-18.5)	50.5 (43.6-57.2)	9.40 (7.61-11.18)	0.036 (0.028-0.043)	0.353 (0.331-0.375)
Model	22.9 (21.8-24.0)	54.1 (52.1-56.3)	14.5 (13.8-15.3)	0.033 (0.031-0.035)	0.231 (0.226-0.236)
65+ years					
Call-takers	32.9 (30.1-35.7)	53.5 (49.4-57.6)	23.7 (21.4-26.0)	0.401 (0.352-0.449)	1.467 (1.373-1.560)
Model	42.8 (41.9-43.7)	66.3 (65.1-67.5)	31.6 (30.8-32.4)	0.290 (0.278-0.303)	1.224 (1.198-1.249)
Male					
Call-takers	30.2 (27.2-33.3)	53.9 (49.1-58.9)	21.0 (18.5-23.5)	0.124 (0.105-0.141)	0.542 (0.506-0.580)
Model	39.0 (38.0-40.1)	63.7 (62.3-65.2)	28.1 (27.3-29.0)	0.097 (0.093-0.102)	0.435 (0.425-0.445)
Female					
Call-takers	21.9 (19.1-24.6)	51.3 (46.0-56.6)	13.9 (12.0-15.8)	0.090 (0.076-0.103)	0.582 (0.547-0.616)
Model	32.4 (31.4-33.4)	62.3 (60.7-63.8)	21.9 (21.1-22.7)	0.069 (0.066-0.073)	0.407 (0.399-0.416)

call-taker (figure 9.1, right). The ensemble model outperformed the individual models regardless of the threshold, except for one that exhibited a slightly better sensitivity at a high FPR exceeding 1.5%. The confusion matrices (table 9.3) illustrate the performance differences in absolute numbers, with the model exhibiting more true positives and fewer false positives than the call-takers.

9.2.3   SEX AND AGE

The model and call-takers exhibited significantly higher PPV and F1-score in men than in women ($p < 0.0001$, independent approximate permutation test) (table 9.2). The model significantly outperformed the call-takers on all metrics for each sex ($p < 0.0001$, paired approximate permutation test). The model performed significantly better in the 65+ group than in the 18-64 year group regarding sensitivity, PPV, and F1-score ($p < 0.0001$, independent approximate permutation test). Similarly, the call-takers performed significantly better in the 65+
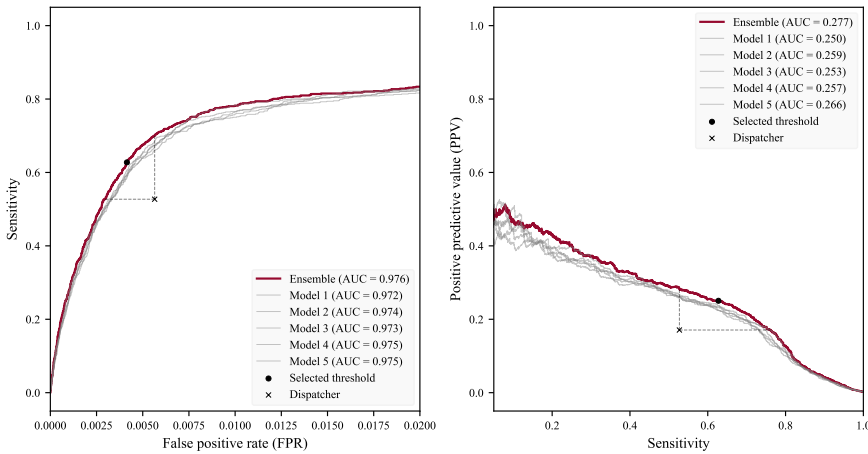
Figure 9.1: Receiver operator characteristic (ROC) curve and PPV-sensitivity curve. Left, the ROC curve and, right, PPV-sensitivity curve (precision-recall curve). Models 1-5 are the individual models that make up the ensemble model.

Table 9.3: Prediction confusion matrices. Confusion matrices of predictions for call takers and the model on the test set. Numbers for the model are given as the rounded mean over eleven runs.

group than in the 18-64 group regarding PPV and F1-score (p < 0.0001, independent approximate permutation test). Finally, the model significantly outperformed the call-takers on all metrics in both age groups (p < 0.0001, paired approximate permutation test).

9.2.4 MODEL EXPLAINABILITY

We performed an occlusion analysis to evaluate the importance of individual words for both positive and negative classifier predictions table 9.4. Among the words with a positive rank score, several words are synonymous with stroke, such as "blood clot", "hemorrhagic stroke", and "stroke". Ambulances are rarely

dispatched because the MH-1813 is not intended for emergencies. Therefore, a word like "ambulance"may also be a strong indicator of call-taker recognition, which the model has learned to mimic. Additionally, most of the remaining words can be linked to stroke-related symptoms such as "double vision", "difficulties speaking", and "hangs". Particularly, words describing the side of the body where symptoms occur ranked high (such as "left", "right", and "side"). Finally, some words were also related to the sudden onset of symptoms (including "suddenly"änd "minutes").

Among the words with a negative rank score, most were strong indicators for specific conditions, symptoms, or body parts that are unrelated to stroke (such as "tetanus", "pregnant", "swollen", "fever", and "the knee"). Another group of words used to describe aspects of treatment that are unlikely to be addressed in a stroke call included "prescription", "bandage", and "OTC". Finally, a small group of words described institutions that are not commonly involved in stroke treatment (such as "psychiatric", "the emergency room", and "the police").

9.3 DISCUSSION

Our results showed that a machine learning framework can substantially improve stroke recognition in medical helpline calls compared to solely relying on human call-takers. This improvement was observed across all performance metrics and for basic patient demographics (age and sex). Our occlusion analysis revealed that the model relied on the relevant predictive features associated with call-taker triaging, patient symptoms, and treatment.

This study does not imply that a machine learning model can replace medical call-takers. The effectiveness of the model is fully reliant on the conversation between the call-taker and caller and the call-taker's ability to skillfully triage the patient. Instead, the model should be used as a supportive tool for call-takers in the decision-making process, contributing to a higher recognition of patients with stroke and potentially boosting the confidence of call-takers in their decisions. A similar machine learning model designed to predict cardiac arrest was tested in a randomized controlled trial (RCT) at CEMS [52]. The results highlighted the necessity of incorporating input from call-takers. The machine learning model for cardiac arrest has subsequently been implemented in daily practice at CEMS, in a setup similar to the one presented in our study. However, implementation of our framework requires further investigation. The relative performance gap between call-takers and the model was larger in our study than in the cardiac arrest study [52], which may affect the results of a potential RCT.

To support future work and discussions beyond the scope of this study, the supplementary material includes the results of a simulation of a live implementation where call-takers are assumed to follow a set of fixed rules based on the

Table 9.4: English translation of words with the largest positive and negative ranking score in calls predicted as stroke and non-stroke, respectively. For this analysis, we used the model with the median F1-score out of 11 randomly seeded runs.

	Positive ranking score		Negative ranking score	
	Stroke predictions, $D = 1,897$		Non-stroke predictions, $D = 342,133$	
	Word, w *(translated)*	Occurrences, $D^{(w)}$	Word, w *(translated)*	Occurrences, $D^{(w)}$
1.	Ambulance	1,680	Tetanus	4,378
2.	Blood clot	895	Pregnant	8,749
3.	Left	1,108	Cut	7,592
4.	Right	1,050	Bandage	4,561
5.	Double vision	84	Amager (a location)	23,776
6.	The words	344	O'clock	94,436
7.	Suddenly	783	The emergency room	42,809
8.	Arm	709	The police	2,903
9.	Side	1,139	Swollen	60,559
10.	Stroke	117	Over the counter (OTC)	4,641
11.	Double	113	The neck	30,151
12.	Control	134	Fever	112,586
13.	Call	39	Prescription	5,450
14.	Numb	94	Centimeter	12,026
15.	Minutes	763	The knee	8,875
16.	Difficulties speaking	44	The pharmacy	10,085
17.	Hemorrhagic stroke	133	The stomach	42,105
18.	Hand	297	Psychiatric	3,688
19.	The ambulance	521	Pneumonia	7,597
20.	Slurred speech	58	Stomach pain	10,551
21.	Blood clots	224	Stool	19,155
22.	Fast	663	The ribs	3,928
23.	Express	44	Bleed	10,501
24.	Blood thinner	259	Bleeding	24,313
25.	Incoherent	15	Ribs	2,941
26.	Lopsided	211	Broken	19,415
27.	Reduced	528	Inflammation	10,050
28.	Hangs	628	Common cold	8,127
29.	Transient	48	Morning or morrow	78,558
30.	Not making sense	14	Swelling	17,762

output of the machine learning framework appendix E. For instance, in one simulation call-takers are assumed to change any stroke negative to a positive, if the model predicts a positive. While the results of the simulation are encouraging, it is important to stress that it is not practically feasible to use a fixed rule set to overrule the call-taker. These results should only be seen as a preliminary indicator of a potential RCT. In practice, a nuanced set of guidelines should be developed over several iterations of implementation and testing.

The performance gap between the model and call-takers could be explained by the rarity of stroke calls to MH-1813 (0.250% of all calls in 2021), which might affect call-taker awareness of stroke as a possible cause of certain symptoms. Additionally, certain stroke symptoms are so rare that some call-takers may never encounter them, increasing the risk of false negatives. The model was trained on more calls than any single call-taker would handle in a lifetime, enabling it to recognize even rare descriptors of stroke. The model is specifically trained to recognize strokes and exclusively learns from actual stroke descriptions, unlike call-takers, who are trained with generalized teaching materials to triage many different conditions. Therefore, call-takers may not have received specific training for patients with stroke and may never have encountered them.

The model performed significantly better on men than on women. This could be attributed to several factors. First, the model may have learned to mimic call-takers with the same bias. Second, women may experience different and more challenging-to-identify symptoms than men [77, 169]. Third, a higher prevalence of male patients with stroke was observed in the training data. Despite these potential sources of bias, the model exhibited less bias than call-takers did. That is, the relative performance improvements were higher for women than for men. This bias could be further reduced using advanced data augmentation and balanced data when training a machine learning model. However, such measures may degrade overall performance.

The improved sensitivity and PPV on the 65+ years group may be explained a higher prior probability of stroke for older patients and stronger evidence from the patient's medical history. The relatively high FOR and FPR for the 65+ group is likely to be a result of the much higher prevalence of stroke cases compared to the 18-64 year olds (0.85% vs. 0.07%). We did not have data to estimate potential bias related to race, ethnicity, language, accent, or dialects. Previous studies on speech recognition for call centers have indeed found that non-native speakers had a higher rate of transcription errors [233]. Since our model was trained on a representative - and therefore unbalanced - sample, we expect it to behave similarly. Future research should look to address these shortcomings, for example by utilizing selfsupervised learning on massive amounts of diverse, unlabeled data covering multiple languages, accents, and dialects.

Due to European data regulations (GDPR), it was not possible to manually transcribe MH-1813 calls to train a new speech recognition model, so we had to rely on an existing solution. This also meant that we could not evaluate the word error rate (WER) of the model. Instead, we used the downstream performance of the text classification model when trained in combination with different speech recognition models to choose the best option. Since the focus of this study is the ability to correctly recognize stroke, and not the performance of the speech recognition model alone, this approach is better suited. Indeed, the WER might

be misleading when choosing a speech recognition model for a specific task. For instance, one model might fail to predict redundant minimal response words (e.g. "uh" and "uhm") and make small inflection errors (e.g. "clot" instead of "clots"), which results in a relatively high WER, while another model only fails to predict rare, specialized words that are highly indicative of stroke (e.g. "hemorrhage" and "thrombolysis"), which results in a relatively low WER.

Although we believe that the proposed machine learning framework can be further improved, several alternatives have already been explored in the preliminary experimental phase. The speech recognition model we used was trained on 1-1-2 calls for a previous project [53], and so, was specialized to a domain very similar to that of MH-1813. We also tested an open-source, multilingual model from OpenAI called Whisper [539], but found that performance degraded slightly compared to the model trained on 1-1-2. We hypothesize that this is due to Whisper's inability to handle the specific noise conditions and recognize words from a specialized medical vocabulary.

For text classification, we used an ensemble of multi-layer perceptrons (MLPs). We also tested convolutional, recurrent, and self-attention (i.e. Transformer) architectures. However, this did not improve performance. In addition, we tested a pre-trained self-supervised model. Although many of these models are freely available to the public, they are primarily trained on English data. Only relatively few options exist for the Danish language, none of which are specialized in the medical domain. We used a monolingual Danish BERT model, which has previously been shown to outperform a multilingual alternative from Google for Danish named entity recognition [289]. However, this also did not result in a significant performance improvement. We hypothesize that the number of ground truth stroke positives was too small for these advanced models to learn more complex patterns than the MLP ensemble. In addition, a self-supervised model would likely benefit from being pre-trained on speech or text data from the target domain. Although training such large-scale foundation models have the potential to improve the classification model further, it is beyond the scope of this study. Thus, we chose the simpler MLP ensemble. We have included references to reviews of self-supervised learning for speech and text in the references [228, 461]. Notably, it is not uncommon for small, simple models to match or outperform large, pre-trained models for text-classification tasks [193].

This study has some limitations. First, the mapping of call recordings to electronic records was incomplete due to technical limitations in the computer-aided dispatch (CAD) registry, which limited the number of calls available to us. Of note, there was no obvious pattern of bias related to the unmapped calls, and we included all calls with matching audio files, regardless of dispatcher performance. The results could potentially be improved if more calls were available for analysis. Second, calls without a call-taker indicated diagnostic category were

not included in the validation and test data because the call-taker's performance could not be evaluated. Moreover, in exploratory analyses, the model performed worse on these calls, which might be attributed to differences in population characteristics (tables 9.1 and 9.2). Finally, the ground truth stroke labelling relied on the patient-reported time of onset being exact; however, estimating the accuracy of the timestamps in DanStroke was impossible.

In conclusion, using the largest collection of audio calls from patients with stroke to date, we developed a machine learning framework that significantly outperformed human call-takers in stroke recognition in medical helpline calls. The framework can assist human call-takers during medical helpline calls. Ideally, this would enable a higher recognition of patients with stroke in the prehospital setting, benefiting both patient outcomes and health service resource allocation.

9.4 METHODS

9.4.1 DATA SOURCES

Copenhagen Emergency Medical Services (CEMS)   The CEMS is responsible for providing prehospital telehealth services in the Capital Region of Denmark, with a catchment area of 1.9 million [143]. CEMS operates two call lines: the 1-1-2 emergency line, similar to 9-1-1 in the United States, intended for acute conditions. The other is the medical helpline 1813 (MH-1813, pronounced "'eighteen-thirteen") intended for non- life-threatening conditions that cannot wait until a general practitioner is available [748].

Call-takers for both lines, who are nurses, paramedics, or physicians, can dispatch ambulances. The condition suspected by the call-taker is categorized based on a predefined diagnostic index and stored in an electronic record using a CAD system. The CAD records are associated with the Danish civil registration number (CPR number) [583] of the patient. The CPR number is a unique identification assigned to all Danish residents. It is used for interactions with health services and registries, enabling cross-referencing of the data sources used in this study. The call audio is recorded and stored separately from the CAD recordings using a telephone system.

Danish Stroke Registry (DanStroke)   All patients with a final diagnosis of stroke or transient ischemic attack admitted to a Danish hospital within 5 days of symptom onset are recorded in the Danish Stroke Registry [313], also known as DanStroke. This record includes the patient-reported time of onset, stroke type (hemorrhagic, ischemic, or transient ischemic attack), and CPR number of the patient. The diagnosis is obtained according to the national guidelines [28], which includes cere-

bral imaging and full diagnostic workup by neurologists. The validity of the Danish stroke registry has been shown to be high [700], and the number of stroke mimics is therefore minimized in our dataset.

Inclusion and ethics   The Danish Data Protection Agency (P-2021-475) approved this study. Danish law did not require approval from the Scientific Ethics Committee because the data were registry-based. CEMS approved the transcription of all calls made to 1-1-2 and MH-1813. All electronic records were anonymized before analysis, and the researchers did not inspect the calls manually.

9.4.2   Study scope

Stroke prevalence in calls made to the MH-1813 is lower than that in calls made to 1-1-2. Patients with stroke may exhibit different symptoms and symptom severity because MH-1813 is meant for low-acuity incidents, leading to reduced recognition. In addition, MH-1813 call-takers dispatch high-priority transport less frequently, which may affect optimal treatment timing. Therefore, we focused on MH-1813 in this study.

9.4.3   Stroke dataset

Cross-referencing data sources   From the CAD medical records, we included all calls that could be matched to a corresponding audio file for 1-1-2 and MH-1813 from 2015 to 2021 for patients older than 18. The CAD records were matched with the telephone call recordings based on the call start, call duration, and call-taker identity. Due to data incompleteness, and the way the audio data is stored, at CEMS, 2,730,199 contacts could not be matched to their corresponding audio file, however, 2,361,178 contacts were successfully matched. We found no obvious pattern in the matched and unmatched calls, and we included all calls with at matching audio file. Next, a call was regarded as a case of ground truth stroke positive when the CPR number in the CAD record matched that of a DanStroke record, and the patient-reported time of onset was close to the call start time. We allowed a window of 72 hours before and 24 hours after the call starts to account for uncertainty in recording stroke onset time. We excluded calls involving subarachnoid hemorrhage cases. Finally, we considered a call to be a call-taker stroke positive when the call-taker selected the stroke diagnostic category during the call and dispatched an ambulance with the appropriate level of response [144]. To ensure that the effect of the machine learning framework was not over-estimated, we excluded calls where diagnostic category had not been registered from the test set. We still reported the population characteristics and model performance of this group of calls to assess potential bias introduced by excluding

them. A data-flow diagram is included in appendix E. The resulting dataset is the largest dataset of audio files from stroke calls collected to date.

Dataset splitting   We reserved all the MH-1813 calls from 2021 for testing. We used stratified sampling to divide the MH-1813 calls from 2015 to 2020 into validation and training subsets. The training subset was further split into five folds which were used for ensemble training. The calls were stratified based on the ground truth stroke label and the presence of a diagnostic category. Calls without diagnostic categories were only included in the training set. The 1-1-2 calls were used only for training; however, calls from 2021 were discarded to avoid temporal overlap with the test period.

9.4.4   Machine learning pipeline

We employed a two-step machine learning pipeline. First, a call was transcribed using the speech recognition model. Second, the transcript was used as input for the text classification model. The final output score was used to classify whether the call concerned a stroke. The pipeline is illustrated in appendix E.

Speech recognition   The call recordings from the CEMS were stored as 8-bit linear pulse-code modulated audio, sampled at 8 kHz. A call was converted into a log-Mel spectrogram before being input into the speech recognition model. This conversion is a commonly used input representation for speech-processing tasks, which facilitates the identification of linguistic content in audio signals. We used a speech recognition model with a neural network architecture [57], consisting of two-dimensional convolutional layers [376] and blocks of bidirectional long short-term memory layers [269] The output is a sequence of probability distributions over characters of the Danish alphabet, which were then converted into a human-readable transcript using a greedy decoder [217].

Text classification   As input for the classification model, each transcript was transformed into a fixed-size bag-of-words vector, which encoded the occurrence of word and character (n-grams) in a fixed vocabulary. The feature selection procedure is detailed in appendix E. The model was constructed as an ensemble [234] of five identical, independently trained models. Each consists of a stack of neural network layers commonly referred to as a multi-layer perceptron [565]. The final layer has a single scalar output and applies a sigmoid nonlinearity to produce an output score between zero and one.

Threshold calibration and ensembling   For each model in the ensemble, we selected the prediction threshold as the harmonic mean of the two thresholds

that respectively ensure sensitivity and PPV equal to that of the call-takers. This simplifies the comparison by ensuring a trade-off between sensitivity and PPV, similar to that of call-takers.

As the threshold differed for each model in the ensemble, computing the ensemble output score as the average output score of the individual models would not be meaningful. Instead, we first subtracted the threshold from the output score in the logit space (before sigmoid nonlinearity) for each model to obtain the same threshold (0.5). Subsequently, we defined the ensemble output score as the average of the centered output scores. The exact equations are provided in (E.1) and (E.2).

Model training   The speech recognition model was trained on 3,811 manually transcribed random calls (173 h) from the CEMS as part of a previous project [53]. These calls exclusively originated from 1- 1-2 between 2015 and 2018, ensuring no overlap with the test data used for the text classification model. The model was trained using a connectionist temporal classification objective [217].

We trained five models for the text classification ensemble using binary cross-entropy after transcribing all calls in the dataset using the speech recognition model. One training fold was used for early stopping using the F1-score, whereas the remaining four folds and 1-1-2 data were used for training. Thus, each model in the ensemble was trained and validated using different datasets. We ran a grid search with 96 different hyperparameter configurations and selected the ensemble model with the best F1-score for the validation set.

9.4.5   Model explainability

We performed an *occlusion analysis* to better understand the predictions of the text classification model. This involved removing all instances of a given word from the input transcript to evaluate its impact on the model output. The word was removed before vectorization, such that all word and character n-grams associated with the word were discarded. Specifically, let $z^{(n,d,w)}$ be the logit output of model n in the ensemble for transcript d when the word w is occluded. For transcript d, we computed the word impact score $i^{(d,w)}$ as the mean difference between the logit before and after occlusion.

$$i^{(d,w)} = \frac{1}{N_d} \sum_{n=1}^{N_d} \left(z^{(n,d)} - z^{(n,d,w)} \right) \quad . \tag{9.1}$$

We used the logit output to compute the impact score because the difference in sigmoid-normalised output is biased towards zero for values close to 0 or 1. To

select words for inspection, we computed a ranking score, $r^{(w)}$, as the sum of the signed squares of the impact:

$$r^{(w)} = \sum_{d=1}^{N} \text{sign}\left(i^{(d,w)}\right) \left(i^{(d,w)}\right)^2 \quad . \tag{9.2}$$

where sign represents the sign function. Squaring $i^{(d,w)}$ favors rare features with a high impact over common features with a low impact.

9.4.6 STATISTICAL ANALYSIS

We report the F1-score, sensitivity, PPV, FOR (equal to 1- negative predictive value), and FPR (equal to 1 - specificity). Due to the unbalanced nature of the dataset, the negative predictive value and specificity were greater than 99% for all cases. We reported FOR and FPR instead because such large numerical values exhibit low relative variance, thereby obfuscating comparisons. Finally, we report the prediction confusion matrices, ROC curve, and PPV-sensitivity curve, commonly known as the precision-recall curve. All results are reported with up to three significant digits.

We present the results with and without 1-1-2 training data, subgroup analyses based on age (18–64/65+) and sex (male/female), and call-takers performance. We also report the model performance on calls without a diagnostic category from the test year 2021 to assess potential data bias. We tested our results for statistical significance using approximate permutation tests. We used one-sided paired approximate permutation tests for model-to-model and model-to-call-taker comparisons when done on the same subset. For comparisons across different subsets (e.g. male vs. female) we used one-sided independent approximate permutation tests. We computed 95% confidence intervals (CIs) using bootstrapping [167, 170]. In our assessment, we accounted for random variation associated with model training by basing the means, tests, and CIs on the predictions of 11 randomly initialized training runs. Statistical significance was defined as a p-value of less than 0.05.

We used the model with the median F1-score out of the 11 runs for the occlusion analysis. We listed the 30 words with the highest positive ranking scores for calls classified as stroke and the 30 words with the highest negative ranking scores for calls classified as non-stroke.

DATA AVAILABILITY

The datasets used to evaluate call-taker performance and to train and evaluate the machine learning framework are legally restricted by Danish patient privacy

and secrecy laws and are therefore, not publicly available. The data can be made available from the publication date but requires a Data Access Agreement, which is examined and approved by the ethics committees that approved this research.[10] For the same reason, the machine learning framework trained in this study is not publicly available; however, instructions on how to train it are included in the main manuscript and the supplementary material.

CODE AVAILABILITY

The source code can be shared using a Creative Commons NC-ND 4.0, an international license, upon reasonable written request to the corresponding author, and requires a data use agreement.

ACKNOWLEDGEMENTS

AUTHOR CONTRIBUTIONS

JW, JDH and LB share lead authorship and contributed equally to the original drafting of the paper, review and editing, data curation, formal analysis, methodology, validation, and conceptualization. JDH and LB contributed to software development and funding acquisition. SNFB contributed to the conceptualization, data curation, funding acquisition, supervision, and writing - review and

---

[10]Danish Data Protection Agency, `https://www.datatilsynet.dk/english`

editing. LM contributed to conceptualization, funding acquisition, project administration, supervision, and writing - review and editing. MS contributed to supervision and writing - review and editing. HC contributed to the methodology, supervision, and writing - review and editing. CK contributed to the conceptualization, funding acquisition, methodology, project administration, supervision, and writing - review and editing. HCC contributed to the conceptualization, data curation, funding acquisition, methodology, project administration, supervision, and writing - review and editing. HCC and CK share last authorship. The three lead authors directly accessed and verified the underlying data reported in this manuscript. JW's affiliations are exclusively academic.

DISCUSSION AND CONCLUSIONS

DISCUSSION

The rapid progress of machine learning that started about a decade ago with the 2012 ImageNet competition and the work of Krizhevsky, Sutskever, and Hinton [357] all but slowed down during the course of this project. Following the papers by Choi, Jang, and Alemi [107], Hendrycks, Mazeika, and Dietterich [254], and Nalisnick et al. [473], the field of out-of-distribution detection saw a surge of interest that has since grown yearly. Similarly, representation learning for speech has continued to advance with notable works such as CPC [499], wav2vec 2.0 [26] and data2vec [24].

Having examined VAEs and self-supervised approaches for representation learning in parts II and III, in this discussion we will consider how VAEs might be enabled to learn representations more competitive with self-supervised methods for downstream tasks (section 10.1). We will also complement the works on OOD detection in part II, which focused on image data, with a discussion about the OOD detection for audio and speech data (section 10.2). Since the work presented in part IV only had limited focus on uncertainty, we will also present and discuss the use of calibration techniques for the stroke recognition model, drawing connections to how such systems might be perceived and used in practice (section 10.3).

10.1 REPRESENTATION LEARNING WITH VARIATIONAL AUTOENCODERS

In this thesis we studied two different approaches to speech representation learning: VAEs in chapters 4 to 7 and self-supervised methods in chapter 6. As we saw in chapters 4 and 5, the probabilistic formulation of VAEs provides benefits for their application to uncertainty quantification, although competitive methods based on self-supervised foundation models have also been successful [47, 255, 715]. Nonetheless, as discussed in chapter 6, self-supervised methods are superior to VAEs when it comes to performance on most downstream tasks, such as speech recognition and spoken language understanding. While this statement of course depends on the task and the amount and type of unlabeled and labeled data, self-supervised methods for speech are generally evaluated on downstream tasks that are harder and more complex than those used to evaluate VAEs. This is evidenced by the leaderboards on benchmarks such as SUPERB, SLUE and ZeroSpeech [164, 603, 723].[11] In the following, we will discuss potential directions

---

[11]One example of comparable evaluation is that between tables 7.1 and A.6 where self-supervised methods can be seen to outperform VAEs for phoneme recognition while self-supervised methods

of future research and try to shed light on why representations from VAEs underperform on downstream tasks and how they might be improved.

10.1.1   GIVING UP?

Two key observations might lead to the pessimistic view that VAEs are fundamentally unsuited for representation learning. First, VAEs are trained to maximize the ELBO which, when tight, is a marginal likelihood objective that is independent of the latent variables $\mathbf{z}$. Second, as we saw in (3.29), the ELBO can be written to reveal that its maximization leads to minimization of the mutual information between the observed and latent variables. In the following we will discuss each of these observations.

Marginal likelihood maximization   For any latent variable model $p_\theta(\mathbf{x}, \mathbf{z})$ we can use the posterior distribution $p_\theta(\mathbf{z}|\mathbf{x})$ to map a given observation $\mathbf{x}$ to its latent representation $\mathbf{z}$. We hope this representation is useful, for instance in the sense that it can improve performance on downstream tasks compared to other methods. Essential to the usefulness is the posterior distribution $p_\theta(\mathbf{z}|\mathbf{x})$.

On the other hand, when we use maximum likelihood to train a latent variable model, we maximize the log-marginal likelihood $\log p(\mathbf{x})$. This can be equivalently phrased as maximizing the KL-divergence of the true data distribution $p_{\text{data}}(\mathbf{x})$ to the learned marginal, $D_{\text{KL}}(p_{\text{data}}(\mathbf{x}) \, \| \, p_\theta(\mathbf{x}))$.[12] The first key observation then is that the marginal $p_\theta(\mathbf{x})$, which we maximize, and the posterior $p_\theta(\mathbf{z}|\mathbf{x})$, which we want to produce useful representations, are separate properties of a latent variable model. Any combination uniquely defines a valid latent variable model. Hence, maximizing the marginal is a useless criterion for representation learning, regardless of the measure of usefulness [7, 288].

---

are additionally reported for word-level speech recognition.

[12]Maximizing the likelihood $\sum_{n=1}^{N} \log p_\theta(\mathbf{x})$ w.r.t. parameters θ is asymptotically equivalent to minimizing $D_{\text{KL}}(p_{\text{data}}(\mathbf{x}) \, \| \, p_\theta(\mathbf{x}))$:

$$
\begin{aligned}
\underset{\theta}{\arg\min}\, D_{\text{KL}}(p_{\text{data}}(\mathbf{x}) \, \| \, p_\theta(\mathbf{x})) &= \underset{\theta}{\arg\min}\, \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\log \frac{p_{\text{data}}(\mathbf{x})}{p_\theta(\mathbf{x})}\right] \\
&= \underset{\theta}{\arg\max}\, \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\log p_\theta(\mathbf{x})\right] \\
&= \underset{\theta}{\arg\max}\, \lim_{n\to\infty} \underbrace{\sum_{n=1}^{N} \log p_\theta(\mathbf{x}_n)}_{\text{log-likelihood}}.
\end{aligned}
$$

The last equality follows from the law of large numbers.

VAEs, however, are not trained to directly maximize the marginal likelihood, but a lower bound of it, the ELBO,

$$\mathcal{L}(\mathbf{x}; \theta, \phi) = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x}|\mathbf{z}) \right]}_{\text{reconstruction loss}} - \underbrace{D_{\text{KL}} \left(q_\phi(\mathbf{z}|\mathbf{x}) \, \| \, p(\mathbf{z}) \right)}_{\text{KL-divergence to prior}} . \qquad (10.1)$$

Since the ELBO is not independent of the latent variables, its maximization will typically depend on the latent variables and the posterior $p(\mathbf{z}|\mathbf{x})$. Nonetheless, being a lower bound of the marginal likelihood suggests it could suffer from similar issues under "right" conditions.

Mutual information minimization   As we saw in section 3.3.5, optimizing the ELBO involves minimizing the mutual information between the latent and observed variables. We reprint equation (3.29) below for reference.

$$\mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})} \left[\mathcal{L}(\mathbf{x}; \theta, \phi) \right] = \underbrace{\mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})} \left[\log p_\theta(\mathbf{x}|\mathbf{z}) \right]}_{\text{average reconstruction}} - \underbrace{D_{\text{KL}} \left[q_\phi(\mathbf{z}) \, \| \, p(\mathbf{z}) \right]}_{\text{marginal KL to prior}} - \underbrace{I_{q_\phi(\mathbf{x},\mathbf{z})} \left[\mathbf{x}; \mathbf{z} \right]}_{\text{mutual information}} .$$

Note that we let $q_\phi(\mathbf{x}, \mathbf{z}) = q_\phi(\mathbf{z}|\mathbf{x})\hat{p}(\mathbf{x})$ for ease of notation. We can see that optimizing the ELBO means learning a good one-to-one mapping from $\mathbf{z}$ to $\mathbf{x}$ (average reconstruction), fitting the aggregated posterior to the prior (marginal KL-divergence), while also not learning any stochastic dependency between $\mathbf{x}$ and $\mathbf{z}$ (mutual information). These objectives are conflicting and especially the minimization of mutual information seems counterproductive to our goal of learning useful representations [645].

How did this ever work?   Even though these properties suggest that VAEs might not work well for representation learning, VAEs have consistently been shown to be able to learn meaningful representations [102, 278, 434, 614, 657]. We can understand the reason for this by noting that both of the above scenarios can only occur if the optimization is sufficiently unconstrained.

Specifically, the maximum marginal likelihood estimation in practice takes place not over all possible models, but over a constrained parametric class defined by the inference network $q_\phi(\mathbf{z}|\mathbf{x}) \approx p(\mathbf{z}|\mathbf{x})$ and the generative model $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})$. This introduces a coupling between the marginal $p_\theta(\mathbf{x})$ and the posterior $p(\mathbf{z}|\mathbf{x})$ which makes the maximum likelihood solution unlikely to correspond to completely useless representations [288]. Particularly successful approaches have made this coupling stronger by sharing parameters between the inference and generative models [102, 434, 614, 657]. Similarly, maximizing the ELBO by exactly minimizing the mutual information term would correspond to

a complete posterior collapse which necessitates a trade-off with the other terms. For a given model, this solution is almost surely a local minimum, and one that can often be avoided in practice.

Following this discussion, we can highlight the main challenges for VAEs to learn representations more competitive with self-supervised methods. We must better understand how to impose constraints on the model class and optimization problem in a structured way in order to 1) ensure a strong coupling between the marginal likelihood and the usefulness of the posterior, and 2) make minimization of the mutual information term very difficult. In the following, we will discuss some possible approaches.

10.1.2   DESIGNING THE LATENT SPACE

Architectural improvements and hierarchies of latent variables are two related, and interesting, approaches to learning more semantic features in VAEs. Some success has been demonstrated in the image domain where models like BIVA [434], NVAE [657], and VD-VAE [102] use up to 78 stochastic layers and have achieved tight likelihoods and high-quality samples. Nonetheless, only limited effort has been put towards evaluating the usefulness for downstream task of representations learned by these very deep models. For speech, a hierarchical model operating on multiple temporal scales, such as the Clockwork VAE [577] adapted for speech in chapter 7, might help better capture dependencies at longer ranges and encode more semantic features. For instance, phonetic content for pronunciation might be learned at lower layers, speaker identity at the upper layers, and semantic features, such as word-meaning, in between. Although some work has successfully separated speaker identity from content by modifying the latent variables and their dependencies [278], models that can learn a deep hierarchy of features for speech remains an open challenge. The work presented in chapter 7 represents an effort to make progress towards such a model, but more work remains.

10.1.3   ADDING A FEW LABELS

Another way to improve the representations learned in VAEs is via semi-supervised learning. Here, a few labels are used to inform which patterns are learned from a large, mostly unlabeled, data set. This is usually done by defining a new stochastic variable as the target and deriving a semi-supervised version of the ELBO that accommodates using the labels when they are available, or marginalizing the target variable when they are not. The VAE is then trained on the labeled and unlabeled data simultaneously.

Although VAEs are strong models for semi-supervised learning [343, 344, 434], self-supervised methods have established themselves as superior for most tasks in this setting [26, 310, 413]. Despite being theoretically appealing, joint objectives for semi-supervised learning in VAEs have practical drawbacks. By training on labeled and unlabeled data simultaneously, semi-supervised VAEs are often less flexible than self-supervised methods which divide the training into two separate phases. By first fitting a general foundation model in an expensive pre-training phase that can later be fine-tuned, self-supervised foundation models can be fine-tuned for several downstream tasks, at relatively low cost. VAEs, on the other hand, must often learn the labeled downstream tasks while simultaneously training on the unlabeled data. This is computationally expensive, but also requires retraining with the unlabeled data, when new supervised data becomes available or a new task is added. Improvements in semi-supervised learning with VAEs might lead to more useful representations, but by requiring labels it remains less attractive than fully unsupervised learning.

10.1.4   Approximating less

VAEs are inherently approximate. Training is performed on a lower bound of the likelihood and inference is variational, amortized and for non-hierarchical models, mean field. As a consequence, the gradient itself is estimated inexactly, usually by only one posterior sample, leading to non-negligible variance. For that reason, it seems plausible that using tighter bounds and reducing gradient variance might indirectly yield more useful representations by improving learning in VAEs.

Importance weighting the ELBO [69] provides a tight bound on the likelihood, reduces gradient variance, and induces a complex implicit posterior distribution. Its use has become standard when reporting likelihood benchmarks but as we described in section 3.3, Rainforth et al. [542] demonstrated that using it during training introduces high gradient variance for the inference network, hurting its ability to learn useful representations. However, Later works largely solved this issue and showed that the reduced variance leads to improved likelihoods [35, 562, 651].

Despite much of this progress being prior to the latest, very large hierarchical VAEs, such as BIVA [434], NVAE [657], and VD-VAE [102], these models are all trained with the single-sample ELBO using regularly reparameterized gradient estimator. Instead, these models rely on advanced inference networks [434] and architectural improvements [102, 657] to overcome the challenges of training hierarchical VAEs. This indicates that there might be untapped potential in consolidating the work on low-variance gradient estimators, advanced architectures and inference methods, which could help improve representation learning

in VAEs.

10.1.5   MASKING

As discussed in chapter 6, masking is one of the driving forces behind the success of self-supervised methods for speech [26, 151]. By removing tokens from the input, or an early feature extraction layer, and tasking a model with inferring their representations, models are forced to learn how neighboring tokens relate to those that are masked. Depending on the size of the mask, these dependencies can be more or less semantic in nature. For instance, the wav2vec 2.0 model is well-known to learn representations that have high similarity with word identity and meaning [513].

In comparison, reconstruction in VAEs is done from latent variables that are inferred from the full, unmasked input. Since all information is generally available, this might allow the encoder and decoder models to perform well for reconstruction, even with no, or limited, use of contextual dependencies. Furthermore, VAE reconstruction almost always targets the direct input in order to learn the distribution over the training data and to enable generating new samples. However, compared to many self-supervised methods that target intermediate, learned representations, this forces VAEs to encode all aspects of the input that are important to its representation in the observed space. As we saw in chapter 4, this will generally include low-level features that are necessary for accurate input reconstruction. Such features require large latent space representations and model capacity [102, 657], but are usually of lesser interest for downstream tasks [26].

Since the ELBO does not immediately allow for masking as part of the training, it has been only sparsely examined for VAEs. Specifically, masking has seen the most attention for VAEs within missing data imputation. In this setting, the input is partially observed, and often represented as a segmentation into observed and missing parts via a mask that indicates where the data is missing. The model is then trained to infer the latent variable from the observed data and reconstruction also deals only with the observed data. By comparison to self-supervised approaches that use masking for representation learning, VAE training in the missing data setting focuses on reconstructing the observed data rather than the missing, which might not lead to the same benefits in representation learning. The idea of using VAEs to impute missing data was already examined in the seminal paper by Rezende, Mohamed, and Wierstra [557]. Here the model was trained with fully observed data and used to impute data in an iterative sampling approach post hoc, leaving the learned representations unchanged. Previous work that trains on partially observed data has largely focused on the ability of these models to yield high-quality imputations within the tabular and image

data domains and have not probed for the effects on the learned latent representation [293, 447].

Including masked objectives into the principled probabilistic framework of VAEs might require rethinking the principles themselves. Nonetheless, relaxing the requirement of exact input reconstruction and enforcing the need for encoding (and decoding) contextual information is one of the strongest candidates to pave the way for future types of VAEs to learn representations that are competitive with self-supervised methods.

10.2 Out-of-distribution detection for speech

Somewhat casually ignoring the fact that this thesis is mainly interested in audio and speech data, the studies of part II dealt with OOD detection only in the context of image classification. The reason for this is that image classification has been the de facto standard task for benchmarking OOD detection methods since the first papers on the topic [107, 254, 347, 473]. In this section, we shall provide some perspective on OOD detection in the context of audio data.

In certain cases, it might be possible to classify an image as OOD based only on the color or intensity of a single pixel. For most distributions, however, we are interested in a test that is sensitive to more semantic properties of the data - for instance the animal class, as discussed in the introduction (section 1.2.2). For speech, this property holds too; a single amplitude value is unlikely to provide much information about the distributional origin of a speech sample. In fact, audio samples are one-dimensional which likely makes them even less discriminative compared to image pixels which generally have multiple channels e.g. for color. Instead, whether a sample is OOD must be inferred from correlations and patterns found in amplitudes along the temporal dimension. In this way, the task of OOD detection has a similar formulation across different data types.

Nevertheless, few studies have explored OOD detection specifically for speech. In one recent study, the authors propose a supervised approach that uses a kNN classifier fitted to representations of the training data [67]. The approach is tuned to a fixed recall on the training data and operates on full length audio samples. This method is similar to image methods that fit a Gaussian mixture to an appropriate model layer [389, 715]. In the early work of Hendrycks and Gimpel [253], the OOD detection method (based on softmax probabilities) is also benchmarked on audio datasets including TIMIT and Aurora-2 [196, 520]. These works strongly suggest that with good data representations, we can use similar methods for OOD detection on both audio and images.

Despite the similarities of the OOD detection task and its methods between data types, audio comes with unique properties. Where images are immediately interpretable to a researcher by visual inspection, audio requires playback, of-

ten in real-time, to reveal its contents. This fact alone is likely to have reduced interest in audio data by simply making it more time-consuming to work with than images. On top of this, audio is generally variable length and requires more computer memory than images which increases complexity and the amount of compute required to obtain equivalent results compared to images. As such, audio-based OOD detection is likely to benefit from progress in standardized frameworks for machine learning like PyTorch [517] and HuggingFace [395, 706] which make it increasingly easier to work with irregular data types.

Another fundamental difference between audio and images is that audio is additive; two signals can be added together to form a new, mixed signal that is meaningful in the sense that it might as well be a true recording. Addition of two images on the other hand produces an image that is not generally representative of what might be photographed. This property of images has led to the practice of evaluating OOD detection performance by comparing examples between different datasets (e.g. MNIST versus FashionMNIST). Interestingly, the additive property of audio allows complementing this dataset-versus-dataset evaluation with an approach where in-distribution examples are augmented to be OOD by mixing with unknown sources. This was initially explored by Hendrycks and Gimpel [253] but has not seen a wide investigation.

Finally, most work on OOD detection has focused on an example-centric approach where an entire example must be classified as either in- or out-of-distribution. For audio, however, it is easy to create examples where local segments are OOD while other parts are not. Using the additive property, we could add a loud Morse signal on top of a dialogue to change a segment of in-distribution audio to OOD. Similarly, silence might be used as a kind of padding to insert entirely OOD segments in between in-distribution parts without addition and without significant boundary effects. For images, the equivalent approach is similar to the successful regularization technique CutMix [731] which pastes unrelated images into an example image. In the context of OOD detection, this approach creates strong boundary errors, which might be used or learned to easily detect such synthetic OOD images. Examples that contain locally OOD parts might occur naturally too; both in the image and audio domains. All kinds of unknown sound can be considered OOD and hence any recording might be locally OOD. In image segmentation, unknown object classes might similarly be seen as locally OOD parts of an example. Future research might examine further how detection of locally OOD examples differs from the dominant example-centric approach.

10.3   UNCERTAINTY OF THE STROKE RECOGNITION CLASSIFIER

Our work on stroke recognition in chapter 9 focuses on the predictive performance of the ensemble model and an analysis of feature importance, but does
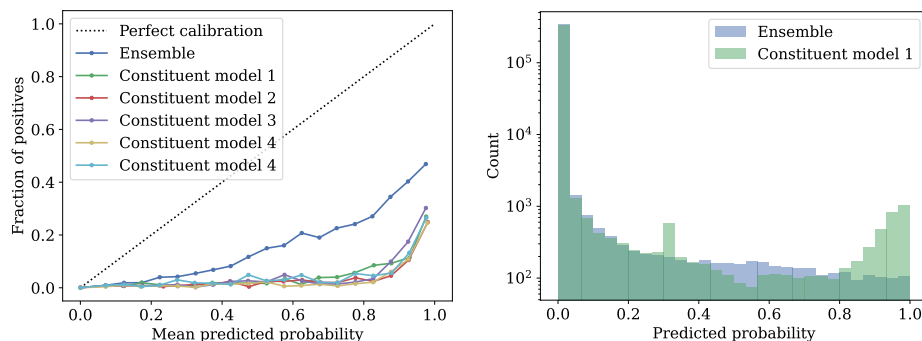
Figure 10.1: Calibration curve for the uncalibrated stroke recognition ensemble (left) and the histogram of predicted probabilities (right) for the test set. We use the ensemble that achieved the median F1-score reported in figure 9.1 and table 9.4.

not explicitly consider uncertainty estimation. As we discussed in section 1.2.1, such a model can be calibrated to predict probabilities that are aligned with the empirical probability of the model being correct on some validation set. Here we present and discuss model calibration for the ensemble model.

We compute the calibration curve by sorting the probabilities predicted on the test set into a number of bins spanning the range from zero to one. For each bin b, we compute the mean predicted probability $\bar{p}$ and the fraction of examples for which the model predicted correctly r_b. The calibration curve is the drawn from the $\{(\bar{p}_b, r_b)\}_b$ pairs. For any given bin, a perfectly calibrated model would have the same fraction of correct predictions as that bin's mean value, $\bar{p}_b = r_b \,\forall\, b$.

The calibration curve for the uncalibrated stroke recognition ensemble and its constituent models is plotted in figure 10.1 along with a histogram of its predicted probabilities. The miscalibration issue that we previously discussed is clearly visible as a strong overconfidence for both ensemble and constituents, although the ensemble is much better calibrated than its constituents. Since the ensemble's output probability is computed as the harmonic mean of the five constituent model probabilities, it can never exceed the maximum probability predicted between the constituent models. This property tends to make ensemble probabilities less extreme and, since the constituent models are overconfident, this results in better calibration.

To calibrate the ensemble model, we can use methods such as Platt-scaling [528] or isotonic regression [733]. In either case, we fit a simple regression model (logistic or isotonic) to the predicted probabilities and the target labels on the
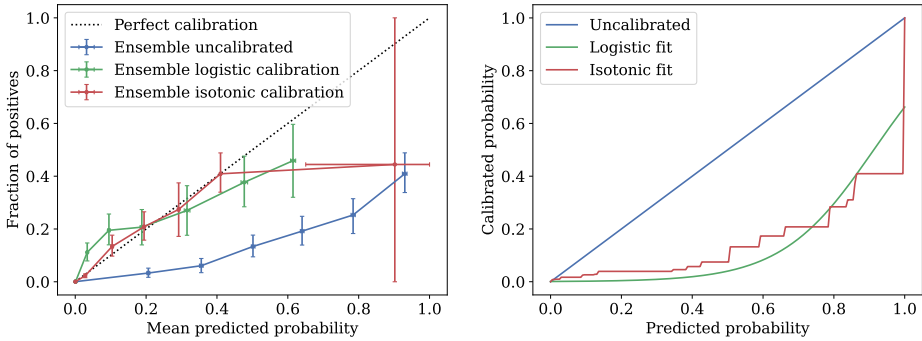
Figure 10.2: Calibration curves using sigmoid and isotonic calibration fits for the stroke recognition ensemble model (left) and the calibration fits (right) for the test set. We use the ensemble that achieved the median F1-score reported in figure 9.1 and table 9.4.

validation set and use it to adjust the probabilities predicted on the test set. We show the resulting calibration curves on the left in figure 10.2 and the logistic and isotonic fits on the right with 95% bootstrap confidence intervals on the bin centers (x error) and fraction of positives (y error). We see that both methods result in quite good calibrations[13] and that the predicted probabilities are shifted towards smaller values. Since stroke cases have low prevalence, high probability is predicted only for a few examples (see the histogram in figure 10.1). This leads to a lack of data for the calibration fits at high predicted probabilities which can be seen to result in poor generalization to the test set, especially for the nonparametric isotonic regression.

Clinicians in intensive care units and emergency departments have been found to strongly agree that a singular focus on overall accuracy cannot alone ensure sustained trust in a model [646]. Clinicians expect an alert to present a prediction that aligns with patient status. Despite expert-agreed thresholds for when alerts should be triggered, however, many alerts may not be aligned since class imbalance and ambiguous information in many predictive problems in healthcare can lead to models with relatively low predictive precision [52, 53, 655, 698]. In turn, this is likely to lead to alarm fatigue [173] and can undermine the sustained use and endorsement by clinicians of such systems [222]. The stroke recognition ensemble presented in chapter 9 is not exempt from this risk. With a precision of 24.9% (95% confidence interval 24.3−25.5%) it will on average be wrong three out

---

[13]Brier scores on test set: Uncalibrated = 0.003500, logistic = 0.001807, isotonic = 0.001774. Relative improvement in Brier score compared to uncalibrated (Brier skill score): Logistic = 0.4830, isotonic = 0.4924.
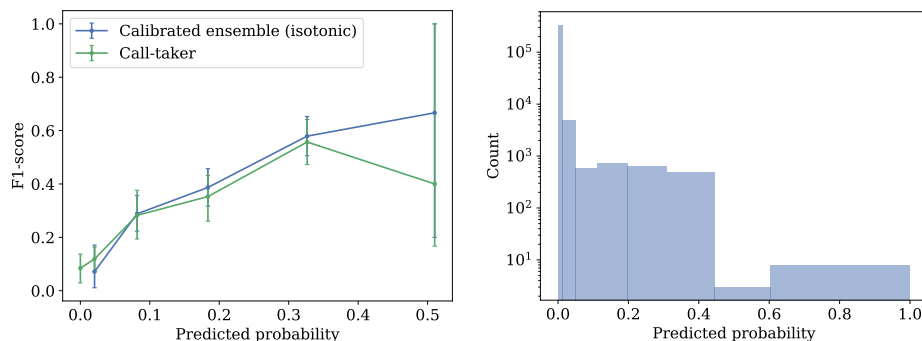
Figure 10.3: Comparison of the F1-score of the stroke recognition ensemble and call-takers. The F1-score is computed on subsets of the test dataset made by binning on the predicted probabilities of the calibrated ensemble. We see that the relative performance improvement of the ensemble over call-takers is higher towards more certain predictions. We use the ensemble that achieved the median F1-score reported in figure 9.1 and table 9.4.

of four times it predicts a stroke. This unfortunately risks alarm fatigue among its potential users, diminishing its effect in practice.

Calibrated probabilities in the alerts presented to users might be a way to alleviate the problem of alarm fatigue. They would allow users to discern between certain and uncertain predictions and also enable the system to present users only with predictions that have a minimum probability of being correct. Similar approaches have been suggested by clinicians and interviews indicate that predictive uncertainty is perceived by experts as a kind of explanation that complements the prediction [646]. In figure 10.3 we show the F1-score of the ensemble model and the call-takers computed on subsets of the data created by binning the calibrated probabilities. We note that, as might be expected, both call-taker and ensemble model performance increase with increased model certainty. This indicates that selecting, based on certainty, which predictions to present to users might indeed help build trust in the system, reduce alarm fatigue, and ensure a practical impact.

CONCLUSIONS AND OUTLOOK

Chapter 1 introduced the motivational cases of automated medical coding and stroke recognition and used them to exemplify the importance of out-of-distribution detection, and, by extension, representation learning. In the context of these cases, we discussed possible machine learning system designs for decision support and considered potential sources of uncertainty and ideal model behavior. The cases form a reference point for the thesis as a whole, connecting its contributions within out-of-distribution detection and representation learning back to practical applications. In this conclusion, we will review the studies presented in previous chapters in the context of the discussion of chapter 10 and the recent progress in the field, and point to interesting directions of future research. As **chapter 2** already details the contributions made by this thesis, we will not reiterate them in detail in this conclusion.

Chapter 3 provided in-depth technical background that could only be covered briefly by the individual studies. We first introduced uncertainty as a concept in the context of information and probability theory (section 3.1). We then defined the task of out-of-distribution detection and reviewed existing work on the problem (section 3.2). Finally, we provided technical background for variational autoencoders (section 3.3).

Chapter 4 showed how hierarchical variational autoencoders can fail at likelihoodbased OOD detection due to an overemphasis on low-level features that generalize between different data distributions. In other words, the latent representations of high-dimensional data from different distributions overlap, especially for latent variables low in the hierarchy. By exploiting that VAEs tend to learn more abstract features at latent variables high in the hierarchy, we were able to define a likelihood-ratio score that focused more on features unlikely to be shared between datasets and performed much better for OOD detection. In line with our discussion in section 10.1, these findings show that VAEs can indeed learn useful latent representations, although good performance, at least for OOD detection, might require selecting an appropriate subset of latent representations to use. Similar variations among features learned at different layers have also been identified for self-supervised models and speech representations [513]. To obtain good downstream task performance, this suggests that, besides seeking to improve representations overall, future research effort should also be directed towards finding good methods for selecting relevant subsets of latent

variables in a given hierarchical VAE for a certain task.

Chapter 5 took a different approach to OOD detection than chapter 4 and focused on developing a model-agnostic method. We showed that by phrasing OOD detection as a statistical testing problem and combining different tests, orthogonal properties of the individual tests could be leveraged to improve the OOD detection performance over any single test. The formulation of OOD detection as a statistical test also allows for better guarantees for such systems in practice. For instance, as also discussed in chapter 5, the statistical framework enables false positive rate control which is a valuable property in many practical applications especially if they involve high-risk actions such as in medical decision support.

Chapter 6 provided an overview of unsupervised neural speech representation learning. Such approaches have recently matched supervised methods on many tasks and represent a significant advance in low-resource settings, such as speech recognition for minority languages. We found that for the purpose of learning good representations in an unsupervised manner, self-supervised learning seems to have better inductive biases, or at least pose a more forgiving learning problem, than do VAEs. As discussed in chapter 6 and section 10.1, this likely relates more to inductive biases imposed by implicit constraints in the optimization problem and architecture than to the underlying formalism. For instance, in discussion about the weaknesses of VAE-based approaches (section 10.1) we concluded that their challenges could not be directly attributed to the maximum marginal likelihood objective. Indeed, the masked pre-training objective widely used for successful self-supervised methods has also been identified to correspond to a maximum marginal likelihood objective [466].

This also leaves potential for future work to improve the ability of VAEs to learn useful representations. As discussed in chapter 6 and section 10.1, promising approaches include adopting masked objectives for VAE training, improving architectural designs to impose better inductive biases, and incorporating advances in gradient estimators more widely [35, 542, 562, 651], especially works on large VAE models [102, 434, 657].

Chapter 7 conducted a comprehensive evaluation of stochastic and deterministic generative models, focusing on their model likelihood. The chapter also introduced a novel hierarchical VAE type model for speech, by drawing inspiration from the Clockwork VAE [577]. Despite the limitations of VAEs for representation learning as compared to self-supervised methods (chapter 10), for sequence data, hierarchical VAE models that operate on multiple temporal scales provide a natural framework for encoding distinct feature categories. For instance, pronunciation features might be learned at lower layers, speaker identity at upper

layers, and semantic features in intermediate layers. Despite successful attempts at isolating speaker identity from content in some existing work [278], developing VAE models with the capacity to learn a deep hierarchy of features for speech persists. The model presented in chapter 7 is an attempt at this challenge.

Chapter 8 examined existing works on automated medical coding and found that in many cases training was suboptimal and evaluation standards were biased. We performed a revised comparison of the selected models and provided updated conclusions on the relative performance of models, and the impact of rare codes and long discharge note documents.

The practical impact of the work of chapter 8, and the work in the field of medical coding in general, depend on a number of factors. Much of the field has focused on the MIMIC datasets which originate from the emergency department and ICU of a single hospital. These datasets have enabled much of the progress in the field, but their singular data source also reduces the generality of the derived results and risks biasing the directions of research deemed most impactful [314, 315, 633, 666]. Furthermore, the complexity and multi-label nature of medical coding has lead to a high prevalence of label errors in MIMIC-III [588]. While difficult to remove entirely, having multiple diverse datasets would also alleviate the risk that the biases of such errors lead to misguided conclusions. Besides gathering more data, improved evaluation methods, such as human-in-the-loop, could be useful to increase the reliability of results.

A common weakness of medical coding models is that performance varies widely between classes. The long-tailed distribution of code-frequency is particularly challenging and leads to underperformance on rare codes. Practical applications could still benefit from such models though. By limiting model predictions to a subset of codes for which they perform well, practitioners could focus on harder to code cases. This selection of cases in practical applications of medical coding could also benefit from research into selective prediction [200]. Although pre-training is an obvious approach to improve performance when little data is available, compared to the effect of pre-training in other domains [26, 151, 161, 404, 461], the improvements observed by using pre-trained models for medical coding have been limited [194, 307, 454, 515, 740]. This suggests an untapped potential for future research into more targeted pre-training for medical coding.

While our work in chapter 8 focused on unimodal medical coding models, it is highly likely that future state-of-the-art models will augment discharge summaries with multi-modal inputs such as medical code descriptions, synonyms, and hierarchies [30, 75, 338, 469, 671, 717, 729]. This is particularly useful since coding standards are updated regularly. The ICD standard for instance sees revisions and new codes added yearly, with local adoption following national guide-

lines [83]. Leveraging such modalities will enable adapting models to updated coding standards without having to gather large amounts of new data.

Chapter 9 studied how machine learning might be used to improve decision-making at emergency services in relation to stroke detection. We saw that a model was able to improve significantly on the stroke recognition ability of call-takers alone and that the features it used were sensible and related to symptoms and descriptions of stroke.

In section 10.3, we discussed how calibrating the predictive uncertainty of the stroke model is likely to be necessary to ensure sustained use of such a model in practice. In figure 10.3, we noted that, as expected, model performance measured by F1-score increased with increasing model certainty which underlines the likely usefulness of uncertainty estimates in better matching practitioner expectations of the predictive performance. Nonetheless, basic metrics of model performance might still be obstacles for its practical usefulness. Specifically, the rarity of stroke cases lead to relatively high false positive rate and low precision, likely to induce alarm fatigue among its users. Similar effects are likely to have influenced the practical impact of a similar system for cardiac arrest detection which also showed significant improvements in a retrospective study [53]. Although the model later matched the retrospective results in a prospective study, it did not ultimately result in improved call-taker performance [52].

Nevertheless, the strong retrospective performance of the stroke recognition model indicates that there is significant potential for augmenting the medical interview to allow better recognizing stroke cases. Possible improvements to the system could include using the audio signal to detect speech-related symptoms such mumbling or slurring, or integrating with electronic health-records to cross-reference with patient history. Even so, directly predicting the diagnosis from the conversation is not the only path towards practical impact. By suggesting informative questions to the medical professional, a system could also help guide the course of conversation to avoid missing important details and keep an overview, and to in turn improve the performance of the model. Ultimately, machine learning has proven capable of contributing meaningfully to medical conversations aiming to improve patient outcomes. Future work seems poised to make a significant positive impact on the healthcare industry over the years to come.

PART VI

APPENDICES

Self-Supervised Speech Representation Learning: A Review

This appendix is a piece of original research by the candidate, previously published elsewhere and reprinted here with permission:

Abstract

Although supervised deep learning has revolutionized speech and audio processing, it has necessitated the building of specialist models for individual tasks and application scenarios. It is likewise difficult to apply this to dialects and languages for which only limited labeled data is available. Self-supervised representation learning methods promise a single universal model that would benefit a wide variety of tasks and domains. Such methods have shown success in natural language processing and computer vision domains, achieving new levels of performance while reducing the number of labels required for many downstream scenarios. Speech representation learning is experiencing similar progress in three main categories: generative, contrastive, and predictive methods. Other approaches rely on multi-modal data for pre-training, mixing text or visual data streams with speech. Although self-supervised speech representation is still a nascent research area, it is closely related to acoustic word embedding and learning with zero lexical resources, both of which have seen active research for many years. This review presents approaches for self-supervised speech representation learning and their connection to other research areas. Since many current methods focus solely on automatic speech recognition as a downstream task, we review recent efforts on benchmarking learned representations to extend the application beyond speech recognition.

Phase 1: Pre-train

Phase 2: Downstream
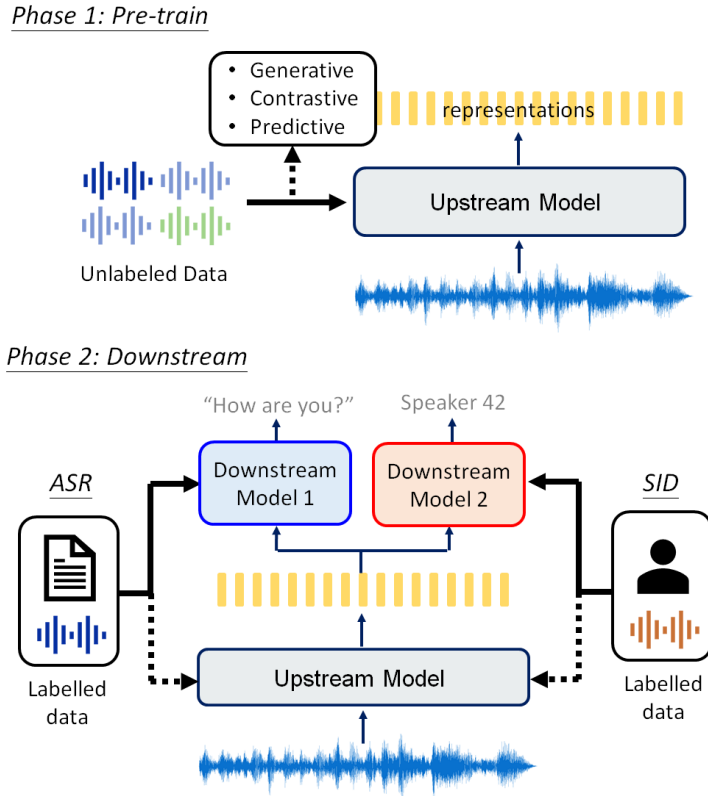
Figure A.1: Framework for using self-supervised representation learning in downstream applications.

A.1 INTRODUCTION

Over the past decade, deep learning approaches have revolutionized speech processing through a giant leap in performance, enabling various real-world applications. Supervised learning of deep neural networks has been the cornerstone of this transformation, offering impressive gains for scenarios rich in labeled data [60, 260, 377]. Paradoxically, this heavy reliance on supervised learning has restricted progress in languages and domains that do not attract the same level of labeling investment.

To overcome the need for labeled data, researchers have explored approaches that use unpaired audio-only data to open up new industrial speech use-cases and low-resource languages [332, 370, 433]. Inspired by how children learn their

first language through listening and interacting with family and surroundings, scientists seek to use raw waveforms and spectral signals to learn speech representations that capture low-level acoustic events, lexical knowledge, all the way to syntactic and semantic information. These learned representations are then used for target downstream applications requiring a minimal number of labeled data [41, 261, 378]. Formally, representation learning refers to algorithms for extracting latent features that capture the underlying explanatory factors for the observed input [41].

Representation learning approaches are generally considered examples of *unsupervised learning*, which refers to the family of machine learning methods that discover naturally occurring patterns in training samples for which there are no pre-assigned labels or scores [318]. The term "unsupervised" is used to distinguish this family of methods from "supervised" approaches, which assign a label to each training sample, and "semi-supervised" approaches, which utilize a small number of training samples with labels to guide learning using a larger volume of unlabeled samples. Examples of unsupervised learning techniques include k-means clustering [218], mixture models [317], autoencoders [264], and non-negative matrix factorization [384]. *Self-supervised learning* (SSL) is a fast-growing subcategory of unsupervised learning approaches, which are techniques that utilize information extracted from the input data itself as the label to learn representations useful for downstream tasks. For example, unsupervised k-means clustering doesn't adhere to this definition of self-supervision since it iteratively minimizes the within-cluster variance during learning. In this review, we focus on self-supervised learning approaches.

Figure A.1 outlines self-supervised representation learning in relation to downstream applications. There are two stages in this framework. In the first stage, we use SSL to pre-train a *representation model*, also called an *upstream model* or a *foundation model*. In the second stage, downstream tasks use either the learned representation from the frozen model, or fine-tune the entire pre-trained model in a supervised phase [263]. Automatic speech recognition (ASR) and speaker identification (SID) are examples of downstream applications in figure A.1.

It is considered desirable for learned speech representations to be disentangled, invariant, and hierarchical. Since spoken utterances contain much richer information than the corresponding text transcriptions—e.g., speaker identity, style, emotion, surrounding noise, and communication channel noise—it is important to learn representations that disentangle these factors of variation. Furthermore, invariance of the learned features to changes in background noise and in the communication channel ensures stability with respect to downstream application scenarios. Learning feature hierarchies at the acoustic, lexical, and semantic levels supports applications with different requirements. For instance, whereas a speaker identification task benefits from a low-level acoustic represen-

tation, a speech translation task requires a more semantic representation of the input utterance.

Due to the popularity of SSL, reviews have been published about the technology in general [55, 176, 418] as well as its application to natural language processing (NLP) [415, 535, 563, 711] and computer vision (CV) [312]. Recently, a brief overview with a general focus on speech representation learning was published [58]. However, none of these overviews focus exclusively on SSL for speech processing. Since the speech signal differs greatly from image and text inputs, many theories and technologies have been developed to address the unique challenges of speech. One review addresses speech representation learning based on deep learning models [373], but does not address recent developments in self-supervised learning. This motivates this overview of speech SSL.

The structure of this paper is arranged as follows. Appendix A.2 briefly reviews the history of speech representation learning, and appendix A.3 reviews current speech SSL models. Appendix A.4 surveys SSL datasets and benchmarks, and discusses and compares results from different works. Appendix A.5 analyzes successful SSL approaches and offers insights into the importance of technological innovations. Appendix A.6 reviews zero-resource downstream tasks that utilize SSL. Finally, appendix A.7 summarizes the paper and suggests future research directions.

A.2   HISTORICAL CONTEXT OF REPRESENTATION LEARNING

In this section we present the historical background of the current surge in self-supervised representation learning methods in the context of two previous waves of research work in the 1990s and 2000s. The discussed approaches go beyond speech to describe the overall landscape of machine learning development during the past few decades.

A.2.1   CLUSTERING AND MIXTURE MODELS

Initial research in learning latent speech and audio representations involved simple models in which the training data likelihood was optimized directly or via the expectation–maximization (EM) algorithm.

Early work used simple clustering methods. For example, in work such as [537, 702], word patterns were clustered semi-automatically using techniques such as k-means, after which isolated words were recognized by finding the training cluster closest to the test data.

Through time, modeling techniques improved such that subword units were represented by Gaussian mixture models (GMMs) [198], which facilitated the modeling of more variability in the input data. GMMs were first built for context-

independent phonemes; state-clustering algorithms [727] then resulted in GMMs for context-dependent phonemes. Each latent component of these mixture models acted as a template of a prototypical speech frame, making it difficult to handle large volumes of data with diverse characteristics. Furthermore, dynamical models like hidden Markov models (HMMs) [38] allowed for the processing of continuous speech rather than just isolated word recognition. These generative GMM and HMM models were trained by maximizing the likelihood of data given the model, which could be accomplished in either an unsupervised or a supervised manner.

Another line of research focused on extracting speech features from generative models. The main objective here was to render the knowledge learned by generative models accessible to discriminative downstream classifiers, or to map variable-length sequences to fixed-length representations. Feature vectors were derived from the parameters of trained GMM models. In the case of *Fisher vectors*, the features were the normalized gradients of the log-likelihood with respect to the model parameters (mixture weights, means, and variances) of the Gaussian mixtures. An extension of this approach (likelihood ratio score space) used the derivative of the log-likelihood ratio of two models, e.g., a background model and a foreground model. Examples of their use in speech processing include speech recognition [609, 665] and speaker recognition [674]. Subsequent techniques in speaker and language verification [145, 146] similarly extracted parameters (concatenated means) from trained background GMMs as representations that were then combined with low-rank projections of speaker/session- or language-specific vectors.

A.2.2 Stacked neural models

More recently, representation learning has seen a shift of focus towards neural models, which, compared to GMMs and HMMs, offer distributed representations with more capacity to model diverse input signals into efficient latent binary codes. Examples of early techniques include restricted Boltzmann machines (RBM) [263], denoising autoencoders [669], noise contrastive estimation (NCE) [229], sparse coding [385, 494, 608], and energy-based methods [544]. Many of these techniques have also been applied to CV and NLP problems, which provided inspiration for their application to speech.

Higher-capacity neural models were achieved by stacking several neural network layers to build progressively higher-level concept representations. However, these deeper networks also increased the training complexities. For example, approximate training methods such as contrastive divergence [266] were a practical technique to streamline RBM training. Furthermore, deep networks had non-convex objective functions, which often resulted in long training times

compared to GMMs, which are trained using full batches instead of mini-batch learning.

A.2.3   LEARNING THROUGH PRETEXT TASK OPTIMIZATION

A more recent trend is learning networks that map the input to desired representations by solving a *pretext* task. Such studies have several characteristics: (1) All layers are trained end-to-end to optimize a single pretext task instead of relying on layer-wise pre-training (2) Past stacked networks typically had only a few layers, but very deep networks with more than ten layers are now common. (3) It is common to evaluate a representation model on a wide range of tasks. For example, in NLP, a representation model is usually assessed on GLUE, which comprises nine tasks [675], whereas in speech, a representation model can be evaluated on SUPERB, which comprises ten tasks [723], as described in detail in appendix A.4.5.

The cornerstone of this third wave is the design of a pretext task, which allows the model to efficiently leverage knowledge from unlabeled data. The pretext task should be challenging enough for the model to learn high-level abstract representations and not be so easy as to encourage the exploitation of low-level shortcuts. Early breakthroughs included end-to-end learning of deep neural architectures via pretext tasks for restoring the true color of black-and-white images [738], joint learning of latent representations and their cluster assignments [79], and the prediction of the relative positions of image patches [158]. Other popular approaches include variational autoencoders (VAEs) [340, 557]. While typical autoencoders learn data representations using unsupervised objectives by reconstructing the input after passing it through an information bottleneck, VAEs estimate a neural model of a probability density function (pdf) that approximates the unknown "true" distribution of the observed data, for which we only have access to independently identically distributed (iid) samples. It is also important to mention dynamical VAEs [205], which is an extension of VAE for sequential data such as speech.

In the SSL context, a pretext task related to autoencoding is to generate an object from its partial information. Such tasks are widely used in NLP, for example, using the previous tokens in a sentence to predict the next token such as in ELMo [527], the GPT series [541], and Megatron [602], or predicting the masked tokens in a sentence such as with the bidirectional encoder representations from Transformers (BERT) series [151, 421]. Another common pretext task in the third wave is contrastive learning [499], in which a model learns to identify a target instance from a set of negative samples. This approach has become especially popular in the CV context [80, 99, 100, 249]. In this survey, we will mainly focus on techniques for pretext task optimization for speech processing, and discuss

these techniques in detail in appendix A.3.

A.2.4   Other related work

A closely related area of research that is not covered in this review is semi-supervised pre-training methods such as pseudo-labeling (that is, self-training). Pseudo-labeling (PL) relies on a supervised teacher model to label a large volume of speech-only data, which is then used to augment the initial labeled data to train a student model [332, 370, 433, 510]. PL has been successful and widely adopted in the speech community since the 1990s. Other proposed variations of PL include augmenting speech-only data with noise to improve robustness, iterating over the PL process to improve teacher labeling quality, and training student models with more parameters than their original teachers to capture the complexities in vastly larger speech-only data [508, 713, 718]. Both SSL and PL leverage unlabeled speech-only data. One distinguishing factor in PL is the utilization of supervised data for a specific task during model pre-training, which limits the model's focus to a single (or at best a few) downstream tasks. SSL, in turn, is an attempt to learn task-agnostic representations to benefit a wide range of tasks.

Transfer learning (TL) is another closely related area of research for pre-training speech models. TL transfers knowledge captured by models trained on one task to different but related tasks [82]. The past few decades have seen active research on TL and its extension to multitask learning for more general representations. Multilingual and cross-lingual supervised models have proven superior in low-resource speech recognition tasks [138]. SSL can be regarded as a type of TL because knowledge learned from pre-training is used for different downstream tasks. This survey paper focuses on SSL, and not all TL technologies for speech. One survey indeed addresses TL for speech processing [38] but does not include current SSL technologies for speech.

A.3   Speech representation learning paradigms

Due to the characteristics of speech, SSL pretext tasks developed for CV and NLP may not directly apply to speech. Below we summarize the characteristics of speech as compared to CV and NLP.

- *Speech is a sequence.* Unlike CV, in which an image usually has a fixed size representation, it is natural to represent a speech utterance as a variable-length sequence. Therefore, pretext tasks developed for CV cannot generally be directly applied to speech.

- *Speech is a long sequence without segment boundaries.* Both text and speech can be represented as sequences. From this viewpoint, it is natural to apply learning approaches developed for text directly to speech. In NLP, morpheme-like tokens are widely used as sequence units in pre-training. The standard BERT takes 512 morpheme-like tokens as input, usually covering a paragraph including several sentences. However, speech signals consist of sound pressure measurements with thousands of samples per second, resulting in sequences much longer than those for text. Even spectral representations which reduce the sequence length can have hundreds of frames per second. Processing such sequences with typical neural network architectures like Transformers can result in problems with running time and memory requirements. One could gather consecutive frames to form shorter segments, but unlike text, there is no obvious segmentation for unlabeled speech.

- *Speech is continuous.* In NLP, it is common to use a pretext task that models a categorical distribution of masked or future inputs. Since text is easily broken down into individual tokens such as words, subwords, or characters, it is straightforward to define a finite vocabulary for such tasks. However, this idea does not apply to speech modeling because speech signals are continuous; in this sense there is no such thing as a speech vocabulary.

- *Speech processing tasks are diverse.* Building generalizable self-supervised representation models for diverse speech processing tasks is challenging. Speech contains rich, hierarchical information, and different speech tasks may require mutually orthogonal information. For example, speech recognition requires a model that extracts content information but ignores speaker information; in contrast, speaker recognition requires a model that extracts speaker information but removes content information. Therefore, it is challenging to define a self-supervised model whose representations are suitable for both speech recognition and speaker recognition. Analogous considerations apply within CV and NLP.

In the sections below, we group modern SSL pretext tasks designed for speech into three main categories: *generative* approaches, *contrastive* approaches and *predictive* approaches. Figure A.2 shows a timeline of the models covered in these sections with each model colored according to our categorization. Table A.1 summarizes model pretext tasks along within the categories.

A.3.1   NOTATION

To efficiently describe the different approaches, we use a simple notation. Models are assumed to consist of functions $f(\cdot)$ and $g(\cdot)$, where $f(\cdot)$ denotes the rep-

resentation model to be used after pre-training and $g(\cdot)$ is an auxiliary module needed only to support the pretext task. For instance, in a classic autoencoder, $f(\cdot)$ would denote the encoder and $g(\cdot)$ the decoder. For more complex models, these functions might consist of several components indicated by sub-indices $f_1(\cdot) \ldots f_N(\cdot)$. As we will see, many self-supervised models use masking, which replaces some parts of the input or a hidden representation by zeros or a learned vector. We use $m(\cdot)$ to denote a function that applies such masking to its input. Similar to $g(\cdot)$, this function is only used during pre-training.

Given an acoustic input $X = \{x_1, x_2, ..., x_T\}$, $f(\cdot)$ outputs a representation $H = \{h_1, h_2, ..., h_T\}$. The input X may be either the raw waveform samples or a sequence of spectral feature vectors. Both are viable options in practice. For simplicity, we do not distinguish between the two in our notation.

While $f(\cdot)$ always takes an acoustic input, the input to $g(\cdot)$ can be either the acoustic signal or another learned representation. Most importantly, $g(\cdot)$ produces an output that is used for the pretext task but is not used by $f(\cdot)$ to produce the representation H. Hence, $g(\cdot)$ can be discarded after pre-training. Finally, $f(\cdot)$ commonly downsamples the temporal dimension, but again, this is not crucial to understand the models, so consider only a single temporal scale $t \in \{1, \ldots, T\}$ for notational convenience.

We use $Q = \{q_1, q_2, ..., q_T\}$ to denote representations that are quantized via codebook learning. Alternatively, discrete representations may take the form of one-hot vectors, or the equivalent integer IDs, which we denote by $C = \{c_1, c_2, ..., c_T\}$. We use a circumflex to denote that, for instance, $\hat{x}_t$ is an approximation of x_t. Finally, we often use a subscript when defining a loss, $\mathcal{L}_i$, to imply that the total loss is computed as a sum over i, unless otherwise stated.

For some models, we will refer to H as a *contextualized* representation which means that each h_t is a function of some, linguistically speaking, long sub-sequence of X spanning at least several phonemes. Usually, h_t depends on the entire input X or all previous timesteps $X_{[1,t]}$. In contrast, a *localized* representation is one that only depends on a short part of the input $X_{[t-u,t+u]}$, where $u \geqslant 0$. The distinction between contextualized and localized may become fuzzy if u is large, however, this is rarely the case.

After pre-training, the representation model $f(\cdot)$ can be fine-tuned for a downstream task directly or used to extract features which are fed to another model, as visualized in figure A.1. It is not uncommon to use the output representation H, but often representations from hidden layers of $f(\cdot)$ are better suited [513].
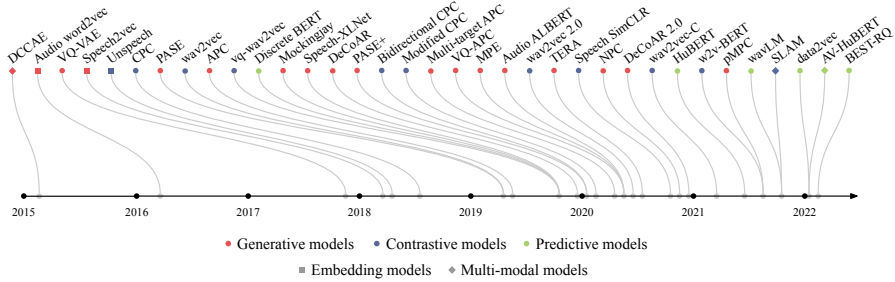
Figure A.2: A selection of models listed according to first publication date on arXiv or conference submission date when this clearly precedes the former. The models are categorized as generative, contrastive, or predictive. In addition, some models are characterized as embedding models or multi-modal models, although most learn frame-level representations from speech only. Some models use a mixture of generative and contrastive tasks. For instance, PASE and PASE+ use a multitask setup, but find that generative tasks are the most important for downstream task performance [516].

A.3.2 GENERATIVE APPROACHES

A.3.2.1 MOTIVATION

In this category, the pretext task is to generate, or reconstruct, the input data based on some limited view. This includes predicting future inputs from past inputs, masked from unmasked, or the original from some other corrupted view. "Generative" as used in this paper hence refers to models that target the original input in their pretext task. Note that this differs from generative models, which learn distributions that allow to sample new data.

A.3.2.2 APPROACHES

Autoencoding Since their introduction in the mid-1990s [264], autoencoders (AEs) have played an essential role in learning distributed latent representations of sensory data. As described above, AEs consist of an encoder and decoder; the pretext task is to reconstruct the given input. The most common type of AE places an information bottleneck on the latent representation by simply having fewer hidden units available than input features. This forces the model to discard low-level details and discourages the learning of trivial solutions. Other models add regularization to the latent space to further improve the quality of

the learned representations. For instance, denoising autoencoders (DAEs) learn latent representations by reconstructing from input corrupted by noise [669]. The Variational Autoencoder (VAE) is a probabilistic version of the AE which defines the latent representation via a posterior distribution over stochastic latent variables [340, 557]. VAEs have been applied to speech in numerous works [3, 126, 188, 277, 278]. The vector-quantized variational autoencoder (VQ-VAE) is another model in this category [500]; it extends the original VAE [340] with a novel parameterization of the posterior distribution for discrete latent representations. The VQ-VAE has been instrumental in generative speech modelling and recent work on generative spoken language modeling has successfully combined the idea of a discrete latent space with self-supervised learning [334, 487, 529].

Specifically, in the VQ-VAE, the continuous representation vector h_t at the output of the encoder is quantized by mapping it to a codebook vector, which is then used as the input to the decoder. This operation is non-differentiable and the gradients of the loss with respect to the encoder parameters must be obtained by approximation. In the VQ-VAE this is done using the straight-through estimator [42], i.e., the gradients with respect to the encoder output are taken to be equal to those with respect to the decoder input (i.e., the quantization step is ignored). Given a learned codebook $A \in \mathbb{R}^{K \times D}$, where K is the codebook size and D is the dimensionality of each codebook vector a_k, the quantized representation q_t of h_t is obtained as

$$q_t = a_k, \text{ where } k = \arg\min_j \left\| h_t - a_j \right\|_2 . \qquad (A.1)$$

The decoder $g(\cdot)$ is an autoregressive model that takes $q_{[1,t]}$ as input to generate x_t [496]. Codebook learning is facilitated by a two-term auxiliary loss similar to classical vector quantization dictionary learning [73, 617]. Gradients for the codebook vectors are given solely by a term that moves codebook vectors a_k closer to the non-quantized vectors h_t. A so-called *commitment term* is added to ensure that non-quantized vectors do not grow unboundedly by enforcing the encoder to keep them close to a codebook vector. This commitment term is optimized only by the encoder. The total VQ-VAE loss for a single timestep is

$$\mathcal{L}_t = \underbrace{\log p(x_t | q_{[1,t]})}_{\text{encoder+decoder}} + \underbrace{\text{MSE}\left(\text{sg}\left[h_t\right], A\right)}_{\text{codebook}} + \underbrace{\alpha \, \text{MSE}\left(h_t, \text{sg}\left[A\right]\right)}_{\text{encoder}} , \qquad (A.2)$$

where $\log p(x_t | q_{[1,t]})$ is a reconstruction likelihood term usually using a categorical distribution, $\text{sg}[x] = x$ is the so-called stop-gradient operator which acts as the identity function during the forward pass but is assumed to have partial derivatives all equal to zero during the backward pass, α is a scalar hyperparam-

eter, and we define $MSE(h_t, A) = \frac{1}{KD} \sum_{k=1}^{K} \sum_{i=1}^{D} (h_{t,i} - a_{k,i})^2$. The loss for a full sequence is the sum or mean over all $\mathcal{L}_t$.

These learned discrete representations have been shown to capture high-level speech information closely related to phonemes, and are useful for applications such as speaker conversion [110]. Vector quantization is not exclusive to VQ-VAE but has seen widespread application within SSL for regularization purposes and to define targets for the pretext task. We will cover these applications below.

The Gumbel softmax [296] is another frequently used approach for obtaining a discrete representation space, and has also been used for AEs [172]. In addition to the approaches discussed above, several other works on speech representation learning take inspiration from the AE framework [20, 21, 322, 555, 593, 736].

Autoregressive prediction   Autoregressive predictive coding (APC) [114, 118] takes inspiration from the classic Linear Predictive Coding (LPC) approach for speech feature extraction [493] and autoregressive language models (LM) for text, where the model learns to predict future information from past. A function $f(\cdot)$ reads the input sequence $X_{[1,t]}$ and outputs a representation sequence $H_{[1,t]}$. The auxiliary module $g(\cdot)$ is a linear projection layer which takes the last vector of $H_{[1,t]}$ as input to approximate x_{t+c}, where $c \geqslant 1$. Thus, c indicates how many timesteps the model predicts ahead. The modules $f(\cdot)$ and $g(\cdot)$ are jointly learned to minimize the L_1 loss between x_{t+c} and its approximation $\hat{x}_{t+c}$. APC is formulated as

$$H_{[1,t]} = f(X_{[1,t]}), \tag{A.3}$$

$$\hat{x}_{t+c} = g(h_t), \tag{A.4}$$

$$\mathcal{L}_t = \|\hat{x}_{t+c} - x_{t+c}\|_1 \ . \tag{A.5}$$

In text-based autoregressive LMs, c is set to 1 to enable autoregressive generation. However, due to the smoothness of the speech signal, neighboring acoustic features are usually similar. Depending on the downstream task, we are often interested in learning so-called *slow features* that typically span multiple input frames [705]. Even the smallest linguistic units of speech—phonemes—span 0.07 seconds on average in the English TIMIT dataset [196], whereas spectrogram frames x_t are typically computed at 0.01 second intervals. Thus, simply predicting the next frame constitutes a trivial pretext task for APC; the original work finds that $c = 3$ performs well. In [115], the APC objective is extended to multi-target training. The new objective generates both past and future frames conditioned on previous context. In VQ-APC [119], quantization is used with the APC objective, which imposes an information bottleneck serving as a regularizer.

A drawback of APC is that it encodes information only from previous timesteps and not the entire input. DeCoAR [406] combines the bidirectionality of the pop-

ular NLP model ELMo [527] and the reconstruction objective of APC to alleviate this issue and allow encoding information from the entire input. It uses a forward LSTM $f_1(\cdot)$ to encode $X_{[1,t]}$ and a backward LSTM $f_2(\cdot)$ to encode $X_{[t+k,T]}$, where $k > 1$:

$$H_{[1,t]} = f_1(X_{[1,t]}) \ , \tag{A.6}$$

$$H'_{[t+k,T]} = f_2(X_{[t+k,T]}) \ , \tag{A.7}$$

$$\hat{X}_{[t+1,t+k-1]} = g(h_t, h'_{t+k}) \ . \tag{A.8}$$

The input feature vector used in the downstream tasks is the concatenation of h_t and h'_t.

Masked reconstruction   Masked reconstruction is largely inspired by the masked language model (MLM) task from BERT [151]. During BERT pre-training, some tokens in the input sentences are masked by randomly replacing them by a learned masking token or another input token. The model learns to reconstruct the masked tokens from the non-masked tokens. Recent work has explored similar pretext tasks for speech representation learning. Similar to the DeCoAR model described above, this allows a model to learn contextualized representations that encode information from the entire input. While we here focus on the models that reconstruct the masked input, it is important to note that masking has also been used extensively for contrastive (appendix A.3.3) and predictive (appendix A.3.4) models.

From a high-level perspective, the training phase of models using masked reconstruction can be formulated as

$$H = f(m(X)) \ , \tag{A.9}$$

$$\hat{x}_t = g(h_t) \ , \tag{A.10}$$

$$\mathcal{L}_t = \|\hat{x}_t - x_t\|_1 \ . \tag{A.11}$$

The exact masking policy defined by $m(\cdot)$ differs from model to model and will be discussed further below. The function $f(\cdot)$ is typically a Transformer encoder [309, 411, 414], but recurrent neural networks have also been used [690]. In general, the Transformer encoder architecture has been adopted widely by self-supervised models for speech within all three surveyed categories. The function $g(\cdot)$ is usually a linear projection or a multilayer perceptron (MLP). Finally, the loss $\mathcal{L}_t$ is commonly computed only for masked timesteps in order to discourage the model from learning an identity mapping.

The masking policies used in NLP can be adapted to speech by considering a speech segment equivalent to a token in a sentence; indeed, the masking strategy of BERT has also been used for speech pre-training [411]. In the standard BERT

masking policy, each token is masked independently at random. However, for speech, masking a single sample or spectrogram frame results in a largely trivial reconstruction task since, as discussed in the paragraph on autoregressive prediction, the smoothness of audio signals may encourage the model to learn to simply interpolate neighboring frames. Therefore, it is common to mask chunks of consecutive frames [311, 411].

We can bring the pretext task closer to the NLP equivalent by using a masking policy where the masked regions of the input correspond to linguistic units. Instead of just masking a fixed number of consecutive frames, pMPC [730] selects masked speech frames according to the phonetic segmentation in an utterance. However, in order to obtain this segmentation, some labeled data is of course needed.

Whereas most studies use masking along the temporal dimension of the input, speech can also be masked along the frequency dimension when spectral input features are used [410, 690]. Frequency masking has been shown to improve representations used for speaker classification [410].

Some studies explore alternatives to masking the input directly. In non-autoregressive predictive coding (NPC) [408], time masking is introduced through masked convolution blocks. Taking inspiration from XLNet [725], it has also been suggested that the input be reconstructed from a shuffled version [615] to address the discrepancy between pre-training and fine-tuning of masking-based approaches.

Regularization methods can further improve on masked reconstruction approaches. DeCoAR 2.0 [405] uses vector quantization, which is shown to improve the learned representations. Furthermore, two dropout regularization methods—attention dropout and layer dropout—are introduced with the TERA model [410, 429]. Both methods are variations on the original dropout method [618].

More generative approaches   Other than the autoregressive and masked reconstruction tasks discussed above, various studies have explored the reconstruction of other targets derived from the input. PASE and PASE+ [516, 549] use multiple targets, including the waveform, log power spectrum, Mel cepstral coefficients (MFCCs), and prosody features. Models that learn acoustic embeddings of small speech segments have targeted future and past spectrogram segments [117, 628, 629], phase information [536], and the temporal gap between two segments [628, 629].

A.3.2.3   Challenges

Although successful NLP models like BERT and GPT are based on generative pretext tasks, the progress have not been translated directly to the speech domain. A speech signal encodes more information than text, such as speaker identity and prosodic features, which makes it harder to generate. However, in order to generate all details of the input, the model must encode all information in the speech signal. Hence, a model that learns to perfectly reconstruct its input may not necessarily have learned to isolate the features of interest and will encode redundant information for a given downstream task.

There are many choices involved in designing a generative pretext task. For instance, masking strategy and the choice of input and target representation (e.g., waveform samples or spectral features). These choices influence what the model learns through the pretext task. However, there is little research on the relationship between task design and the information encoded in the learned representations.

A.3.3   Contrastive approaches

A.3.3.1   Motivation

As discussed above, speech contains many entangled features. Thus, learning to reconstruct the raw speech signal might not be the best way to discover contextualized latent factors of variations. Contrastive models learn representations by distinguishing a target sample (positive) from distractor samples (negatives)

---

[14] $X = \{x_1, x_2, ..., x_T\}$ is the input sequence in which x_t can be an acoustic feature vector (e.g., MFCC, filter bank, or spectrogram features) or a waveform sample. $X_{[t_1:t_2]}$ represents $\{x_{t_1}, x_{t_1+1}, ..., x_{t_2}\}$. $X_{-[t_1:t_2]}$ represents X in which the segment $X_{[t_1:t_2]} = \{x_{t_1}, x_{t_1+1}, ..., x_{t_2}\}$ is masked. x_t^i represents the i-th dimension of x_t. If x_t is a frame in a spectrogram, then the i-th dimension corresponds to a specific frequency bin. $X^{-[f, f+j]}$ refers to a spectrogram X which is masked along the frequency axis from the f-th to $(f + j)$-th bin. We indicate random temporal permutation of a sequence by indexing it with the set $\mathcal{P}_t \equiv \text{PERMUTE}([0, t])$, where $\text{PERMUTE}(\cdot)$ returns a permutation of the given list. We indicate data augmentation (e.g., reverberation) by the function $\text{AUGMENT}(\cdot)$. Subscripts indicate different augmentations. Z represents a localized latent representation sequence of X. $Z^{(l)}$ is Z at the l-th layer of the model used to compute it. $\bar{H}$ is the contextualized sequence H obtained from an exponential moving average (EMA) of the model undergoing training with no masking applied. Q represents a sequence of quantized learned representations, and C is a sequence of discrete cluster IDs. For contrastive models, we specify only positive targets.

[15] PASE uses multiple pretext tasks, but the authors find that reconstruction is most important.

[16] wav2vec-C adds reconstruction loss to wav2vec 2.0.

[17] Discrete BERT obtains codes C from vq-wav2vec.

[18] HuBERT is trained first using cluster IDs of the MFCCs as target and subsequently clusters IDs of the model representations from the last iteration.

[19] WavLM simulates noisy/overlapped speech as inputs.

[20] BEST-RQ obtains codes C by quantizing acoustic features using a random projection quantizer.

Table A.1: An overview of approaches within the three categories of self-supervised learning. Column (a) lists the names of the models and related references, column (b) defines the model input, column (c) defines any corruption of the input or hidden representation, and column (d) defines the target of the pretext task; the pretext task itself is described by the overall model category and the main text. Notation details are in footnote 14.

Model (a)	**Input** (b)	**Corruption** (c)	**Target** (d)
	GENERATIVE MODELS		
Audio Word2vec [122], VQ-VAE [500]	X	-	X
Speech2Vec [117], Audio2Vec [628] - skip-gram	$X_{[t_1,t_2]}$	-	$X_{[t_0,t_1]}, X_{[t_2,t_3]}$
Speech2Vec [117], Audio2Vec [628] - cbow	$X_{[t_0,t_1]}, X_{[t_2,t_3]}$	-	$X_{[t_1,t_2]}$
PASE [516], PASE+ [549][15]	X	-	Different modalities of X
APC [118, 119]	$X_{[1,t]}$	-	$x_{t+c}, c \geqslant 1$
Speech-XLNet [615]	$X_{\mathcal{P}_t}$		$x_{i\text{-}\mathcal{P}_t^c}$
DeCoAR [406]	$X_{[1,t-1]}, X_{[t+k+1,T]}$	-	$X_{[t,t+k]}$
Mockingjay [411], Audio ALBERT [101], DeCoAR 2.0 [405]	$X_{-[t,t+k]}$		$X_{[t,t+k]}$
TERA [410], BMR [690]	$X_{-[t,t+k]}^{-[f,f+j]}$		X
pMPC [730]	$X_{-[t,t+k']}$ ($X_{[t,t+k']}$ is a phoneme)		$X_{[t,t+k']}$
MPE [414]	X	$Z_{-[t,t+k]}$	Z
NPC [408]	X	$Z_{-[t,t+k]}$	X
	CONTRASTIVE MODELS		
Unspeech [459]	$X_{[t_1,t_2]}$	-	$X_{[t_0,t_1]}, X_{[t_2,t_3]}$
CPC [499], wav2vec [584], Modified CPC [558]	$X_{[1,t]}$	-	$z_{t+c}, c \geqslant 1$
Bidirectional CPC [331]	$X_{[1,t]}$ or $X_{[t,T]}$	-	z_{t+c} or $z_{t-c}, c \geqslant 1$
vq-wav2vec [25]	$X_{[1,t]}$	-	$q_{t+c}, c \geqslant 1$
wav2vec 2.0 [26], wav2vec-C [570][16]	X	$Z_{-[t,t+k]}$	$Q_{[t,t+k]}$
w2v-BERT [123]	X	$Z_{-[t,t+k]}$	$Q_{[t,t+k]}$ and $C_{[t,t+k]}$
Speech SimCLR [310][14]	AUGMENT$_1$(X) and AUGMENT$_2$(X)		AUGMENT$_2$(Z) and AUGMENT$_1$(Z)
	PREDICTIVE MODELS		
Discrete BERT [22, 25] [17]	$C_{-[t,t+k]}$		$C_{[t,t+k]}$
HuBERT [275][18], WavLM [98][19]	X	$Z_{-[t,t+k]}$	$C_{[t,t+k]}$
data2vec [24]	X	$Z_{-[t,t+k]}$	$\sum_l \bar{H}_{[t,t+k]}^{(1)}$
BEST-RQ [104][20]	$X_{-[t,t+k]}$		$C_{[t,t+k]}$

given an *anchor representation*. The pretext task is to maximize latent space similarity between the anchor and positive samples while minimizing the similarity between the anchor and negative samples. This approach has been used extensively in the general ML community [586].

A.3.3.2 Approaches

Contrastive predictive coding Contrastive Predictive Coding (CPC) [499] is a prominent example of a contrastive model. CPC uses a convolutional module $f_1(\cdot)$ to produce localized representations z_t with a recurrent module $f_2(\cdot)$ on top that outputs a contextualized representation h_t. An anchor representation $\hat{z}_{t,k}$ is obtained via a linear projection $g_k(\cdot)$ of h_t. The positives and negatives are sampled from the localized representation Z. Hence, at a single timestep t, CPC forms multiple anchor representations $\hat{z}_{t,k}$ for $k \in \{1, \ldots, K\}$ and associates with each one a single positive sample at the corresponding timestep, z_{t+k}, k steps in the future:

$$z_t = f_1(X_{[t-u,t+u]}) \; , \tag{A.12}$$

$$H_{[1,t]} = f_2(Z_{[1,t]}) \; , \tag{A.13}$$

$$\hat{z}_{t,k} = g_k(h_t) \; . \tag{A.14}$$

Each z_t only encodes information from a limited receptive field, while $f_2(\cdot)$ is limited to condition each h_t on previous timesteps $Z_{[1,t]}$. Without these restrictions, the model could collapse to a trivial solution. g_k is a unique transformation per offset k (e.g., a linear projection). The loss function measures the similarity between the anchor representation $\hat{z}_{t,k}$ and the positive z_{t+k} normalized by the total similarity to the positive and negatives. The approach is similar to previous work on Noise-Contrastive Estimation (NCE) [229]. Minimizing the loss corresponds to maximizing a lower bound on the mutual information between h_t and z_{t+k} (and in turn $x_{t+k-u:t+k+u}$) and is hence called InfoNCE:

$$\mathcal{L}_{t,k} = -\log\left(\frac{\exp(\hat{z}_{t,k}^\mathsf{T} z_{t+k})}{\sum_{i \in \mathcal{I}} \exp(\hat{z}_{t,k}^\mathsf{T} z_i)}\right) \; . \tag{A.15}$$

Here, $\mathcal{I}$ is a random subset of N indices which includes the target index $t + k$ and $N - 1$ negative samples drawn from a proposal distribution, e.g., a uniform distribution over $\{1, \ldots, T\}$. Including the target index in $\mathcal{I}$ ensures that the loss is a proper categorical cross-entropy and that minimizing it has the previously stated relation to mutual information maximization. This corresponds to sampling negatives from the same sequence and has been shown to give good performance for phoneme classification [499]. The loss is indexed by k to show that CPC targets

multiple offsets using different projection layers $g_k(\cdot)$. The authors find K = 12 to work well for phoneme classification.

The wav2vec model [584] extends the CPC approach and uses fully convolutional parameterizations for the modules $f_1(\cdot)$ and $f_2(\cdot)$ with receptive fields of 30 ms and 210 ms, respectively. While the CPC loss solves a 1-of-N classification task per (t, k), either assigning the anchor to the positive class or (wrongly) to one of the $N − 1$ negative classes, the wav2vec loss considers a sequence of N independent binary classifications. That is, the anchor is compared independently to the positive and each negative, and the loss is computed as a sum of the associated log-probabilities,

$$\mathcal{L}_{t,k} = -\log(\sigma(\hat{z}_{t,k}^\mathsf{T} z_{t+k})) + \sum_{i\in\mathcal{I}} \log(1 - \sigma(\hat{z}_{t,k}^\mathsf{T} z_i)) \ . \tag{A.16}$$

Here, $\sigma(x) = 1/(1+\exp(-x))$ is the sigmoid function, $\sigma(\hat{z}_{t,k}^\mathsf{T} z_{t+k})$ is the probability of the anchor being the positive sample and $\sigma(\hat{z}_{t,k}^\mathsf{T} z_i)$ is the probability of the anchor being the negative sample. Evidently and contrary to CPC, $\mathcal{I}$ must not include the target index $t + k$ as this would cancel out the positive term.

wav2vec 2.0   The wav2vec 2.0 model combines contrastive learning with masking. As the CPC model, it uses the InfoNCE loss [499] to maximize the similarity between a contextualized representation and a localized representation. However, instead of using the z_t directly as positive and negatives, it uses a quantization module $g(\cdot)$ to obtain a discrete representation. This has the practical implication that one can avoid sampling negatives from the same category as the positive. The model takes as input a waveform and uses a convolutional module $f_1(\cdot)$ followed by a Transformer encoder $f_2(\cdot)$. Masking is applied to the output of the convolutional module:

$$z_t = f_1(X_{[t-u,t+u]}) \ , \tag{A.17}$$
$$H = f_2(m(Z)) \ , \tag{A.18}$$
$$q_t = g(z_t) \ . \tag{A.19}$$

The quantization module $g(\cdot)$ uses a Gumbel softmax [296] with a straight-through estimator. Since the quality of the learned representations is contingent on the quality of the quantization, wav2vec 2.0 combines two techniques to learn high-quality codebooks. First, wav2vec 2.0 concatenates quantized representations from multiple codebooks at each timestep, so-called Product Quantization (PQ) [305]. Also, the primary training loss described below is augmented with an auxiliary term designed to encourage equal use of all codebook entries.

In wav2vec 2.0, anchors are taken to be h_t at masked timesteps only, the positive sample is chosen as the quantized vector, q_t, at the same timestep, and

negatives are sampled from other masked timesteps. The loss is

$$\mathcal{L}_t = -\log\left(\frac{\exp(S_c(h_t, q_t))}{\sum_{i \in \mathcal{I}} \exp(S_c(h_t, q_i))}\right) \ , \tag{A.20}$$

where $S_c(\cdot)$ is the cosine similarity and $\mathcal{I}$ contains the target index t and negative indices sampled from other masked timesteps.

The wav2vec 2.0 approach was the first to reach single-digit word error rate (WER) on LibriSpeech using only the low-resource LibriLight subsets for fine-tuning a pre-trained model (see appendix A.4.2). It has subsequently inspired many follow-up studies. The wav2vec-C [570] approach extends wav2vec 2.0 with a consistency term in the loss that aims to reconstruct the input features from the learned quantized representations, similar to VQ-VAE [551].

A.3.3.3 CHALLENGES

Although representations learned using contrastive approaches have proved effective across a wide range of downstream applications, they face many challenges when applied to speech data. One challenging aspect is that the strategy used to define positive and negative samples can also indirectly impose invariances on the learned representations. For example, sampling negatives exclusively from the same utterance as the positive biases the features towards speaker invariance, which may or may not be desired for downstream applications. Another standing challenge is that since speech input does not have explicit segmentation of acoustic units, the negative and positive samples do not represent a whole unit of language but rather partial or multiple units, depending on the span covered by each sample. Finally, since speech input is smooth and lacks natural segmentation, it can be difficult to define a contrastive sampling strategy that is guaranteed to provide samples that always relate to the anchor as truly positives and negatives in a sound way.

A.3.4 PREDICTIVE APPROACHES

A.3.4.1 MOTIVATION

Similar to the contrastive approaches discussed above, predictive approaches are defined by using a learned target for the pretext task. However, unlike the contrastive approaches, they do not employ a contrastive loss and instead use a loss function such as squared error and cross-entropy. Whereas a contrastive loss discourages the model from learning a trivial solution by the use of negative samples, this must be circumvented differently for predictive methods. For this reason, predictive methods compute the targets outside the model's computational

graph; usually with a completely separate model. Thus, the predictive setup is somewhat akin to teacher-student training. The first predictive approaches were motivated by the success of BERT-like methods in NLP [151] as well as the Deep-Cluster method in CV [79].

A.3.4.2 APPROACHES

Discrete BERT   Applying BERT-type training directly to speech input is not possible due to its continuous nature. The Discrete BERT approach [22] uses a pretrained vq-wav2vec model to derive a discrete vocabulary [25]. The vq-wav2vec model is similar to wav2vec mentioned in the paragraph on contrastive predictive coding but uses quantization to learn discrete representations. Specifically, discrete units c_t are first extracted with the vq-wav2vec model $f_1(\cdot)$ and then used as inputs *and* targets in a standard BERT model $f_2(\cdot)$ with a softmax normalized output layer $g(\cdot)$,

$$c_t = f_1(X_{[t-u,t+u]}) \ , \tag{A.21}$$
$$H = f_2(m(C)) \ , \tag{A.22}$$
$$\hat{c}_t = g(h_t) \ . \tag{A.23}$$

Similar to BERT, the model can then be trained with a categorical cross-entropy loss,

$$\mathcal{L} = \sum_{t \in \mathcal{M}} -\log p(c_t \mid X) \ , \tag{A.24}$$

where $\mathcal{M}$ is the set of all masked timesteps. During training, only the BERT model's parameters are updated, while the vq-wav2vec model parameters are frozen. Discrete BERT was the first model to demonstrate the effectiveness of self-supervised speech representation learning by achieving a WER of 25% on the standard test-other subset using a 10-minute fine-tuning set, setting the direction for many approaches to follow.

HuBERT   Rather than relying on an advanced representation learning model for discretizing continuous inputs, as Discrete BERT, the Hidden Unit BERT (Hu-BERT) approach [275] uses quantized MFCC features as targets learned with classic k-means. Thus, to compute the targets, the k-means model $g_1(\cdot)$ assigns a cluster center to each timestep. Different from Discrete BERT, HuBERT takes the raw waveform as input, rather than discrete units. This helps to prevent loss of any relevant information due to input quantization. HuBERT uses an architecture similar to that of wav2vec 2.0, with a convolutional module $f_1(\cdot)$ and a

Transformer encoder $f_2(\cdot)$, as well as a softmax normalized output layer $g_2(\cdot)$:

$$c_t = g_1(X_{[t-w,t+w]}) \ , \tag{A.25}$$

$$z_t = f_1(X_{[t-u,t+u]}) \ , \tag{A.26}$$

$$H = f_2(m(Z)) \ , \tag{A.27}$$

$$\hat{c}_t = g_2(h_t) \ , \tag{A.28}$$

where w defines the window size used to compute the MFCCs. The categorical cross-entropy loss is computed on both masked, $\mathcal{L}_m$, and unmasked, $\mathcal{L}_u$, timesteps:

$$\mathcal{L}_m = \sum_{t \in \mathcal{M}} -\log p(c_t \mid X) \ , \tag{A.29}$$

$$\mathcal{L} = \beta\mathcal{L}_m + (1-\beta)\mathcal{L}_u \ . \tag{A.30}$$

Again, $\mathcal{M}$ is the set of all masked timesteps, β is a scalar hyperparameter and $\mathcal{L}_u$ is computed as $\mathcal{L}_m$ but summing over $t \notin \mathcal{M}$.

Intuitively, the HuBERT model is forced to learn both an acoustic and a language model. First, the model needs to learn a meaningful continuous latent representation for unmasked timesteps which are mapped to discrete units, similar to a classical frame-based acoustic modeling problem. Second, similar to other masked pre-training approaches, the model needs to capture long-range temporal dependencies to make correct predictions for masked timesteps.

One crucial insight motivating this work is the importance of consistency of the targets which enables the model to focus on modeling the sequential structure of the input. Importantly though, for HuBERT, pre-training is a two-step procedure. The first iteration is described above. Once completed, a second iteration of pre-training follows. Here, representations from a hidden layer of the model from the first iteration are clustered with k-means to obtain new targets c_t.

For HuBERT, only two iterations are needed to match or outperform the previous state-of-the-art results for low-resource speech recognition. And combining the HuBERT approach with the wav2vec 2.0 approach, the w2v-BERT model has managed to improve results even further [123].

WavLM   WavLM emphasizes spoken content modeling and speaker identity preservation [98]. It is largely identical to HuBERT, but introduces two useful extensions.

First, it extends the Transformer self-attention mechanism with a so-called *gated relative position bias*. The bias is added prior to the softmax normalization of the attention weights. For the attention weight at i, j, the bias is computed

based on the input to the Transformer layer at the current timestep i and also incorporates a relative positional embedding for $i - j$. The authors find that this extension improves performance on phoneme and speech recognition tasks.

Second, it uses an utterance mixing strategy where signals from different speakers are combined to augment the training data. Specifically, random sub-sequences from other examples in the same batch are scaled and added to each input example. Only the targets corresponding to the original example are predicted during pre-training. Thus, the model learns to filter out the added overlapping speech.

Most SSL methods are trained on data where each example only contains speech from a single person; therefore, they can perform subpar on multispeaker tasks like speaker separation and diarization.

The WavLM model achieved substantial improvements on the speech separation, speaker verification and speaker diarization tasks in the SUPERB benchmark, while also performing well on many other tasks compared to HuBERT and wav2vec 2.0.

data2vec   Motivated by the success of using an exponential moving average (EMA) teacher for self-supervised visual representations [81, 221], the data2vec model [24] computes targets Y using an EMA of its own parameters. The targets are constructed by averaging hidden representations of the top k layers of the EMA teacher network applied to unmasked inputs. Here, we denote this jointly as $\bar{f}_2(\cdot)$.

The data2vec model was proposed for different data modalities, but for audio it uses an architecture similar to wav2vec 2.0 and HuBERT with a convolutional module $f_1(\cdot)$, a Transformer $f_2(\cdot)$ and masking applied to the Transformer input.

$$z_t = f_1(X_{[t-u,t+u]}) \ , \tag{A.31}$$

$$H = f_2(m(Z)) \ , \tag{A.32}$$

$$Y = \bar{f}_2(Z) \ . \tag{A.33}$$

The teacher network $\bar{f}_2(\cdot)$ is a copy of the Transformer of the student network but with the parameters at training step i, $\theta_{\text{teacher},i}$, given by an EMA of the student parameters over all previous training steps.

$$\theta_{\text{teacher},i} = \begin{cases} \theta_{\text{student},0} & i = 0 \\ \gamma\theta_{\text{student},i} + (1 - \gamma)\theta_{\text{teacher},i-1} & i > 0 \end{cases} , \tag{A.34}$$

where $\theta_{\text{student},i}$ are the parameters of the student network at training step i, updated via gradient descent, and γ is the EMA decay rate.

The data2vec model uses a regression loss between target and prediction. Specifically, to reduce sensitivity to outliers and prevent exploding gradients, it uses the smoothed L_1 loss [206],

$$\mathcal{L}_t = \begin{cases} \frac{1}{2}(y_t - h_t)^2/\eta, & |y_t - h_t| \leqslant \eta \\ |y_t - h_t| - \frac{1}{2}\eta, & \text{otherwise} \end{cases}, \tag{A.35}$$

where the hyperparameter η controls the transition from a squared loss to an L_1 loss.

The data2vec approach was shown to work well for representation learning with either speech, images or text data. It is the first approach to achieve competitive results when trained on any one of the three modalities.

A.3.4.3   Challenges

The iterative nature of pre-training for the HuBERT and wavLM could present a practical inconvenience when working with large volumes of data. Another challenge for these models centers around the quality of the initial vocabulary generated from MFCC features. The data2vec approach improves over other predictive models by allowing the targets to improve continuously via the EMA teacher network; however, student-teacher approaches inflate the existing computational challenges of very large models and may necessitate the use of methods that decrease instantaneous memory utilization such as mixed precision training, model parallelism and model sharding [517].

A.3.5   Learning from multi-modal data

A.3.5.1   Motivation

Multiple modalities are useful in many settings, where each modality provides information that is complementary to other modalities. Multi-modal work includes supervised settings, such as audiovisual ASR [526, 530] and person identification [4] which have been studied for decades. In this section, we focus only on unsupervised representation learning from multi-modal data.

One of the motivations for learning from multiple modalities is that it can reduce the effect of noise, since noise in different modalities is likely to be largely independent or uncorrelated. In addition, learning from speech data with accompanying signals such as images or video can help learn representations that encode more semantic information. Such "groundingßignals can contain supplementary information that can be used by models to infer the content of the speech. Human language learning provides a proof of concept for this, as it is believed that infants benefit from the visual modality when learning language [390].

Early computational models of multi-modal language learning were motivated by (and tried to emulate) human learning of language in the context of the visual surroundings [568].

A.3.5.2 APPROACHES

We define two broad classes of approaches in this area. Specifically, depending on what type of multi-modal data is involved we refer to "intrinsicör "extrinsic"modalities.

Intrinsic modalities are modalities produced directly by the speech source. Examples of intrinsic modalities (besides the speech audio) include images or video of the speaker's face [125, 382], lip-movement [598], articulatory flesh point measurements [699, 708], or simultaneous magnetic resonance imaging (MRI) scans [474]. Typically, learning from multiple intrinsic modalities is done so as to improve robustness to noise, since acoustic noise is likely to be uncorrelated with the other modality(ies). This type of representation learning is often referred to as "multi-view learning"because the multiple intrinsic modalities can be seen as multiple views of the same content. Some typical approaches in this category include

- Multi-view autoencoders and variations [19, 483],

- Multi-modal deep Boltzmann machines [619],

- Canonical correlation analysis (CCA) [272] and its nonlinear extensions [8, 18, 365, 366, 449, 453, 688, 689, 691],

- Multi-view contrastive losses [257, 285],

- More recently, audiovisual extensions of masked prediction methods [598, 599], specifically Audiovisual HuBERT (AV-HuBERT) [598].

Extrinsic modalities are modalities that are not produced by the same source but still provide context for each other. A typical example is an image and its spoken caption: The image tells us what the speech is likely describing, so a representation model that takes both modalities into account will hopefully encode more of the meaning of the speech than a single-modality model. There has recently been a surge of datasets collected for this purpose, usually consisting of images and spoken captions, the audio and image frames in a video, or video clips with their spoken descriptions. A recent review of datasets, as well as methods, in this category is provided by Chrupała [111].

Typical approaches involve learning a neural representation model for each modality, with a multi-modal contrastive loss that encourages paired examples

in the two modalities to have similar representations while unpaired examples remain different [239, 242, 450, 521, 566, 624]

Other choices include training with a masked margin softmax loss [291, 575] or a masked prediction loss [85]. Such models are typically evaluated on cross-modal retrieval, although some work has also used the models for other downstream tasks such as the ZeroSpeech and SUPERB benchmark tasks [522]. Analyses of such models have found that, despite the very high-level learning objective of matching speech with a corresponding image (or other contextual modality), such models often learn multiple levels of linguistic representations from the shallowest to the deepest model layers [112, 240, 578]. They are also able to learn word-like units [241, 523, 685] and can be used for cross-lingual retrieval, by considering the visual signal as an "interlingua-[238, 243, 325]. In some settings, even in the presence of some amount of textual supervision (i.e., the speech is transcribed), visual grounding still helps learn a better representation for retrieval [512].

There has also been growing interest in learning joint speech and text representations using paired and unpaired data. The SLAM approach [31] is an example where speech and text are first represented using two separate pre-trained encoders followed by a multi-modal encoder to build the joint representations. The entire model is trained using a multitask loss including two supervised and two self-supervised tasks.

A.3.5.3 Challenges

One of the challenges of using multi-modal approaches is that the multi-modal data they rely on is often in shorter supply than single-modality data. In addition, multi-modal data is typically drawn from specific domains, for example domains involving descriptions of visual scenes. It is not clear how well the learned speech representations apply to other speech domains that are not necessarily describing or situated in a visual scene, and this question requires further study.

A.3.6 Acoustic word embeddings

Most of the representation learning techniques discussed in the preceding sections are aimed at learning frame-level representations. For some purposes, however, it may be useful to explicitly represent longer spans of speech audio of arbitrary duration, such as phone, word, or phrase-level segments. For example, searching within a corpus of recorded speech for segments that match a given (written or spoken) query can be seen as finding segments whose representations are most similar to that of the query [91, 122, 393, 594]; word embeddings can be defined by pooling representations of instances of a given word [117]; unsupervised segmentation and spoken term discovery can be seen as a problem of

detecting and clustering segments [323, 324]; and even ASR can be viewed as the problem of matching written word representations to representations of audio spans [40, 437, 593].

Several lines of work have begun to address the problem of learning representations of spans of speech, especially word segments, typically referred to as *acoustic word embeddings*. Early work on unsupervised acoustic word embeddings defined them as vectors of distances from the target segment to a number of pre-defined "templateßegments [392]. Later work used variants of neural autoencoders [122, 270, 321, 524]. These are often evaluated on word discrimination, that is the task of determining whether two word segments correspond to the same word or not [78]. This task can be thought of as a proxy for query-by-example search, since the basic operation in search is to determine whether a segment in the search database matches a query segment, and has been used for evaluation of both frame-level (e.g., [322]) and word-level [327, 392] representations.

Since most work on acoustic word embeddings preceded the very recent wave of new self-supervised frame-level representations, one question is whether word (or more generally segment) embeddings could be derived more simply by pooling self-supervised frame-level representations, as has been done for text span embeddings by pooling over word embeddings [647, 687]. Some initial results suggest that at least very simple pooling approaches like downsampling and mean or max pooling are not successful [524, 662], but more work is needed to reach conclusive results.

A.4   BENCHMARKS FOR SELF-SUPERVISED LEARNING

The previous sections presented various methodologies by which to learn speech representations from unlabeled corpora. This section surveys the datasets available to learn and evaluate these representations. We also summarize several studies and their results to demonstrate the usefulness of the learned representations for various downstream tasks.

A.4.1   DATASETS ONLY FOR PRE-TRAINING

Table A.2 summarizes datasets used for pre-training SSL techniques in the literature. These datasets are usually large but with limited or no labels. LibriLight (LL) [320], one of these datasets, is derived from audiobooks that are part of the LibriVox[21] project. LL contains 60k hours of spoken English audio tagged with SNR, speaker ID, and genre descriptions. The speech examples in Audioset [202],

---

[21]https://librivox.org/

which consists of over 2M 10-second YouTube video clips human-annotated with 632 audio events, have also been used for pre-training. Audioset has 2.5k hours of audio of varying quality, different languages, and sometimes multiple sound sources. AVSpeech [175] is another large-scale audiovisual dataset used in SSL research, comprising 4.7k hours of clips from a wide variety of languages. Each clip contains a visible face and audible sound originating from a single speaker without interfering background signals. The 3100-hour audio part of AVSpeech has been used to learn audio-only representations [331]. The Fisher corpus [127] collects over 2k hours of conversational telephone speech, 1k hours of which is utilized for pre-training [311]. Industrial researchers have also begun to build large-scale datasets for learning speech representations. For instance, 10k hours of real-world far-field English voice commands for self-supervised pre-training have been collected at Amazon [570].

In addition to these English and multilingual efforts, researchers have also collected corpora for pre-training Chinese speech representations. Didi Dictation and Didi Call Center [309, 311] are two internal datasets containing respectively 10k hours of read speech collected from mobile dictation application and 10k hours of spontaneous phone calls between users and customer service staff.

A.4.2   Datasets for both pre-training and evaluation

Several datasets that provide both speech and associated transcripts and speaker labels have also been used to develop SSL techniques by enabling in-domain pre-training and evaluation. Such datasets are also listed in table A.2. One of the most commonly used datasets in this category is LibriSpeech (LS) [506], a labeled corpus containing 960 hours of read English speech, which is also derived from an open-source audiobooks project.[21] The corpus consists of subsets *train-clean-100*, *train-clean-360*, *train-other-500*, *dev-clean*, *dev-other*, *test-clean*, and *test-other* used for training, development, and testing, respectively. Subsets tagged with *other* are more challenging utterances from speakers that yield higher WER as measured with previously built models. LS is used for unsupervised representation pre-training by ignoring its labels, and can also be utilized to evaluate the performance of representation on ASR, phoneme recognition (PR), phoneme classification (PC), and speaker identification (SID) tasks. Wall Street Journal (WSJ) [519] is another widely adopted, labeled corpus for pre-training. Its labels can evaluate performance for ASR, PR, PC, and SID. The original WSJ corpus contains 400 hours of English read speech data, and today its *si284* (81 hours), *dev93*, *eval92* subsets are the most-used partitions for unsupervised training, development, and test, respectively. The *si84* (15 hours) partition is also used for training.

The speech community also utilizes multilingual corpora. These are often

large-scale, which facilitates pre-training, but are also partially labeled for ASR evaluation (PC and PR can be enabled via phone-level forced alignment). These corpora include Common Voice (CV) [10], Multilingual LibriSpeech (MLS) [531], VoxPopuli (VP) [676], and BABEL (BBL) [192]. CV is an open-source, multi-language, growing dataset of voices containing 11k hours of audio from 76 languages as of the date this review was written (Common Voice corpus 7.0). Researchers usually use part of this for pre-training (e.g., 7k hours/60 languages in [17] and 430 hours/29 languages in [331]) or evaluation. MLS derives content from read audiobooks of LibriVox and contains data in eight European languages for a total of 50k hours of audio. VP comprises a total of 400k hours of parliamentary speech from the European Parliament in 23 European languages. The entire dataset [17] or a 24k-hour portion [97, 98] thereof has been used for pre-training. BBL consists of 1k hours of conversational telephone speech in 17 African and Asian languages.

Several datasets, including GigaSpeech [92], TED-LIUM 3 (TED3) [258], TED-LIUM 2 (TED2) [567], Switchboard (SWB) [208], TIMIT [196], and VoxLingua107 [659], are labeled and conventionally used for evaluation, while their audio streams are also aggregated to build diversified and large-scale corpora for unsupervised pre-training [17, 331, 615]. GigaSpeech is a multi-domain English ASR corpus with 33k hours of audio collected from audiobooks, podcasts, and YouTube. A subset of 10k audio is transcribed. TED2 comes with 118 hours of English speech extracted from TED conference talks and its transcription for evaluating ASR. Its recordings are clear but with some reverberation. TED3 is an extension of TED2 and comprises 450 hours of talks. SWB is a 260-hour conversational speech recognition dataset containing two-sided telephone conversations. The TIMIT corpus was designed to provide read speech data and its word and phone-level transcriptions for acoustic-phonetic studies. It contains recordings in American English. Compared to the previous corpora labeled for ASR evaluation, VoxLingua107 consists of 6.6k hours of audio in 107 languages and is annotated for language identification. Beyond the original purpose of evaluation, these corpora are also used in pre-training to improve the generalizability of learned representations.

For the purpose of pre-training and evaluating Mandarin speech representations, the authors of [309, 311] also compiled Open Mandarin, an open-source Mandarin dataset of 1.5k hours of speech from the Linguistic Data Consortium (LDC) and OpenSLR.[22] Open Mandarin consists of the HKUST Mandarin Telephone Speech Corpus (HKUST, 200 hours of spontaneous speech, of which 168 hours of audio is used for pre-training; the development and test sets are excluded.) [420], AISHELL-1 [66] (178 hours of read speech), aidatatang 200zh (200 hours, read speech) [37], MAGICDATA Mandarin Chinese Read Speech Corpus (755 hours, read speech) [442], Free ST Chinese Mandarin Corpus (ST-CMDS,

---

[22]https://openslr.org

100 hours, read speech) [623], and Primewords Chinese Corpus Set 1 (100 hours, read speech) [532]. Both HKUST and AISHELL-1 are labeled and are suitable for ASR evaluation.

A.4.3 Datasets for evaluation

Besides the aforementioned datasets, conventional speech processing benchmarks are also used to evaluate self-supervised representations. Studies leverage Hub5, DIRHA, and CHiME-5 to measure the efficacy of representations in ASR. The Hub5 evaluation (LDC2002T43 and LDC2002S09, also referred to as the NIST 2000 Hub5 English evaluation set) contains 40 transcribed English telephone conversations only for testing, where 20 are from conversations collected in SWB studies but not released with the SWB dataset, and the rest are from CallHome American English Speech (LDC97S42). DIRHA [548], short for Distant-speech Interaction for Robust Home Applications, is a database composed of utterances sampled from WSJ, speech of keywords and commands, and phonetically-rich sentences. These utterances are read by UK and US English speakers and recorded with microphone arrays. CHiME-5 [32] is a challenge that aims to advance robust ASR and presents a dataset of natural conversational speech collected under a dinner party scenario with microphone arrays. A team at Amazon Alexa also recorded and transcribed a corpus of 1k hours of audio for model training and evaluation [570].

Researchers also evaluate representations for sentiment analysis with the IN-TERFACE [274] and MOSEI (CMU Multimodal Opinion Sentiment and Emotion Intensity) [732] datasets. INTERFACE is an emotional speech database for Slovenian, English, Spanish, and French, and contains six emotions: anger, sadness, joy, fear, disgust, and surprise, plus neutral. MOSEI is composed of sentence-level sentiment annotations of 65 hours of YouTube videos using emotion categories similar to INTERFACE, but replacing joy with happiness.

In addition, datasets employed to demonstrate the benefit of SSL representations on various tasks include VCTK [664] and VoxCeleb1 [470] for SID/ASV (automatic speaker verification) tasks, FSC (Fluent Speech Commands) [427] for IC (intent classification), QUESST (QUESST 2014) [9] for QbE (query by example), LS En-Fr [350] and CoVoST-2 [677] for ST (speech translation), and ALFFA and OpenSLR-multi for multilingual ASR. The VCTK corpus includes speech data with 109 English speakers of various accents, each reading out about 400 sentences sampled from newspapers. VoxCeleb1 is an audiovisual dataset comprised of short YouTube clips containing human speech. It consists of 1251 unique speakers and 352 hours of audio. FSC contains utterances of spoken English commands that one might use for a smart home or virtual assistant, and is used to evaluate the performance of a spoken language understanding system. The

QUESST search dataset comprises spoken documents and queries in 6 languages to measure the capability of models in spotting spoken keywords from documents. LS En-Fr is a dataset augmenting existing LS monolingual utterances with corresponding French translations to train and evaluate English-French machine translators. CoVoST-2 is a multilingual speech translation benchmark based on CV. It provides data for translating from English into 15 languages and from 21 languages into English, and has a total of 2.9k hours of speech. The ALFFA project[23] collects speech of African languages to promote the development of speech technologies in Africa, and [331] leverages four African languages collected in the project for evaluation: Amharic [626], Fongbe [369], Swahili [201], and Wolof [197]. In the same work [331], the authors further select 21 phonetically diverse languages from OpenSLR to evaluate the generalizability of SSL representations across languages. We denote the collection as OpenSLR-multi below.

Last, [628] puts together five datasets (MUSAN [611], Bird Audio Detection [621], Speech Commands [693], Spoken Language Identification [504], and TUT Urban Acoustic Scenes 2018 [451]) plus an SID task built with the LS *train-clean-100* subset to evaluate the capability of representations on audio event detection. [536] employs the NSynth dataset [174] on top of the six for benchmarking. As many of the datasets are built for research in audio processing, we here provide only a list of these datasets for reference.

---

[23]http://alffa.imag.fr

[24] In table A.2, we use the following abbreviations: **EN**: English; **Multi**: multilingual; **ZH**: Chinese; **Fr**: French; **ASR**: automatic speech recognition; **PR**: phoneme recognition; **PC**: phoneme classification; **SID**: speaker identification; **ASV**: automatic speaker verification; **Sentiment**: sentiment analysis; **ST**: speech translation; **QbE**: query by example or spoken term detection; **IC**: intent classification; **AED**: audio event detection; and **LID**: language identification. We distinguish **PR** from **PC** based on whether the inference is made at the phone level sequentially or the frame level separately. **SID** and **ASV** both evaluate model capability in encoding speaker information; **SID** classifies one utterance into a pre-defined set of speaker labels, whereas **ASV** infers whether a given pair of utterances was uttered by the same speaker.

[25]https://dirha.fbk.eu/node/107

[26]https://chimechallenge.github.io/chime6/download.html

[27]https://github.com/A2Zadeh/CMU-MultimodalSDK/blob/master/LICENSE.txt

[28] The **pre-training corpus**, **Fine-tuning**, and **Test** columns list the datasets used in each work, and the **Task** column lists the tasks performed in the corresponding papers. The **Transfer** column indicates whether the SSL technique is evaluated by its capability for transfer learning, i.e., different datasets are utilized for pre-training and fine-tuning. The **Fine-tuning labels used** column summarizes the amount of labeled examples used in downstream fine-tuning.

Table A.2: Summary of datasets used for pre-training (denoted as PT) or evaluation (denoted as EV) of SSL techniques in the literature. The languages and sizes of the datasets are provided in columns 3 and 4. Column 5 lists the tasks each dataset is used to evaluate. Abbreviations are listed in footnote 24.

Dataset	Purpose	Lang.	Size [hours]	Task	License
LibriLight (LL)	PT	EN	60k	-	MIT License
AudioSet	PT	Multi	2.5k	-	CC BY 4.0
AVSpeech	PT	Multi	3.1k	-	CC BY 4.0
Fisher	PT	EN	2k/1k [311]	-	Linguistic Data Consortium (LDC)
Alexa-10k	PT	EN	10k	-	Not released
Didi Callcenter	PT	ZH	10k	-	Not released
Didi Dictation	PT	ZH	10k	-	Not released
LibriSpeech (LS)	PT/EV	EN	960	ASR/PR/PC/SID	CC BY 4.0
Wall Street Journal (WSJ)	PT/EV	EN	81	ASR/PR/PC/SID	Linguistic Data Consortium (LDC)
Common Voice (CV-dataset)	PT/EV	Multi	11k/7k [17]/430 [331]	ASR/PR/PC	CC0
Multilingual LS (MLS)	PT/EV	Multi	50k	ASR	CC BY 4.0
VoxPopuli (VP)	PT/EV	Multi	400k [17]/24k [97, 98]	ASR	CC0
BABEL (BBL)	PT/EV	Multi	1k	ASR	IARPA Babel Agreement
GigaSpeech	PT/EV	EN	40k/10k [97, 98]	ASR	Apache-2.0 License
TED-LIUM 3 (TED3)	PT/EV	EN	450	ASR	CC BY-NC-ND 3.0
TED-LIUM 2 (TED2)	PT/EV	EN	118	ASR	CC BY-NC-ND 3.0
Switchboard (SWB)	PT/EV	EN	260	ASR	Linguistic Data Consortium (LDC)
TIMIT	PT/EV	EN	4	ASR/PR/PC	Linguistic Data Consortium (LDC)
VoxLingua107	PT/EV	Multi	6.6k	LID	CC BY 4.0
Open Mandarin	PT/EV	ZH	1.5k	ASR	CC BY-NC-ND 4.0, Apache License v.2.0, Linguistic Data Consortium (LDC)
HKUST	PT/EV	ZH	168/200	ASR	Linguistic Data Consortium (LDC)
AISHELL-1	PT/EV	ZH	178	ASR	Apache License v.2.0
Hub5'00	EV	EN	13	ASR	Linguistic Data Consortium (LDC)
DIRHA	EV	EN	11	ASR	See link for details[25]
CHiME-5	EV	EN	50	ASR	See link for details[26]
Alexa-eval	EV	EN	1k	ASR	Not released
INTERFACE	EV	Multi	16	Sentiment	No information
MOSEI	EV	EN	65	Sentiment	See link for details[27]
VCTK	EV	EN	44	SID/ASV	CC BY 4.0
VoxCeleb1	EV	Multi	352	SID/ASV	CC BY 4.0
Fluent Speech Commands	EV	EN	14.7	IC	CC BY-NC-ND 4.0
QUESST 2014 (QUESST)	EV	Multi	23	QbE	No information
LS En-Fr	EV	En-Fr	236	ST	CC BY 4.0
CoVoST-2	EV	Multi	2.9k	ST	CC0
ALFFA	EV	Multi	5.2–18.3	ASR-multi	MIT License
OpenSLR-multi	EV	Multi	4.4–265.9	ASR-multi	CC BY-SA 3.0 US, CC BY-SA 4.0, CC BY 4.0, CC BY-NC-ND 4.0, Apache License v.2.0
AED datasets	EV	-	-	AED	CC BY 4.0 (MUSAN, Speech Commands, NSynth, Bird Audio Detection), CC0 (Spoken Language Identification), Non-Commercial (TUT)

[29] Train/test split made available by [499] on Google drive https://drive.google.com/drive/folders/1BhJ2umKH3whguxMwifaKtSra0TgAbtfb.

[30] Utilizes official training or test split.

[31] English utterances used in experiments. The utterances correspond to approximately 3 hours for training, 40 minutes for development, and 30 minutes for testing.

[32] The 6 AED datasets used in [628] are MUSAN [611], Bird Audio Detection [621], Speech Commands [693], Spoken Language Identification [504], TUT Urban Acoustic Scenes 2018 [451] plus an SID task built with LS *train-clean-100*. In addition to the 6 datasets, [536] use the NSynth dataset [174] for evaluation.

[33] A collection of AudioSet, AVSpeech, CV-dataset, LS, WSJ, TIMIT, Speech Accent Archive

Table A.3: A summary of common experiment settings for various SSL evaluations (Part 1). Networks are usually pre-trained with SSL techniques, augmented with prediction heads, and fine-tuned (or trained) with labeled data in downstream tasks for benchmarking. We follow the abbreviations introduced in table A.2. Column descriptions are in footnote 28.

Work	pre-training corpus	Task	Dataset		Transfer	Fine-tuning labels used
			Fine-tuning	Test		
CPC [499]	LS 100 hrs	PC	LS 100 hrs[29]	LS 100 hrs[29]	-	80[34] hrs
		SID	LS 100 hrs[29]	LS 100 hrs[29]	-	80[34] hrs
PASE[516]	LS 50 hrs [547]	SID	VCTK[30]	VCTK[30]	✓	44 hrs
		Sentiment	INTERFACE[31]	INTERFACE[31]	✓	3 hrs
		PR	TIMIT[30]	TIMIT[30]	✓	4 hrs
		ASR	DIRHA[30]	DIRHA[30]	✓	11 hrs
Audio2Vec [628]	AudioSet	AED	6 AED datasets[32]	6 AED datasets[32]	✓	See [628] for details
APC [114, 118]	LS 360 hrs	ASR	WSJ si284[36]	WSJ dev93	✓	72 hrs
		ST	LS En-Fr[30]	LS En-Fr[30]	-	236 hrs
		SID	WSJ si284[35]	WSJ si284[35]	✓	65[34] hrs
wav2vec [584]	LS 80/960 hrs, LS 960 hrs + WSJ si284	ASR	WSJ si284	WSJ eval92	✓	81 hrs
		PR	TIMIT[30]	TIMIT[30]	✓	4 hr
PhasePredict [536]	AudioSet	AED	7 AED datasets[32]	7 AED datasets[32]	✓	See [536] for details
Bidir-CPC[331]	LS 960 hrs, CPC-8k[33]	ASR	WSJ si284, LS 960 hrs, TED3[30]	WSJ eval92, LS test-clean, LS test-other, TED3[30], SWB[30]	Different datasets for training and test	81/960/450 hrs
		ASR-multi	ALFFA[30]	ALFFA[30]	✓	4 languages, 5.2–18.3 hrs
		ASR-multi	OpenSLR-multi[30]	OpenSLR-multi[30]	✓	21 languages, 4.4–265.9 hrs
MockingJay [411]	LS 360 hrs	PC	LS 360 hrs	LS test-clean	-	0.36[34] /1.8[34] /3.6[34] /18[34] /45[34] /360 hrs
		SID	LS 100 hrs[37]	LS 100 hrs[37]	-	90[34] hrs
		Sentiment	MOSEI[30]	MOSEI[30]	✓	65 hrs
CPC modified [558]	LS 100 hrs	PC	LS 100 hrs[29]	LS 100 hrs[29]	-	80[34] hrs
	LS 100 hrs, LS 960 hrs, LL 60k hrs	PC	CV-dataset[30]	CV-dataset[30]	✓	1 hrs
vq-wav2vec [25]	LS 960 hrs	ASR	WSJ si284	WSJ eval92	✓	81 hrs
		PR	TIMIT[30]	TIMIT[30]	✓	4 hrs
DeCoAR [406]	LS 100/360/ 460/960 hrs, WSJ si284	ASR	WSJ si284	WSJ eval92	-	25[34]/40[34]/81 hrs
		ASR	LS 100/360/ 460/960 hrs	LS test-clean, LS test-other	-	100/360/460/960 hrs
MT-APC[115]	LS 360 hrs	ASR	WSJ si284[36]	WSJ dev93	✓	72 hrs
		ST	LS En-Fr[30]	LS En-Fr[30]	-	236 hrs
PASE+[549]	LS 50 hrs [547]	PR	TIMIT[30]	TIMIT[30]	✓	4 hrs
		ASR	DIRHA[30]	DIRHA[30]	✓	11 hrs
		ASR	CHiME-5[30]	CHiME-5[30]	✓	50 hrs
AALBERT [101]	LS 360 hrs	PC	LS 100 hrs[35]	LS 100 hrs[35]	-	80[34] hrs
		SID	LS 360 hrs[35]	LS 360 hrs[35]	-	288[34] hrs

Table A.4: A summary of common experiment settings for various SSL evaluations (Part 2). See the caption of table A.3 for a detailed description of all the abbreviations used in this table.

Work	pre-training corpus	Task	Dataset Fine-tuning	Test	Transfer	Fine-tuning labels used
BMR [690]	WSJ si284, LS 960 hrs	ASR	WSJ si284	WSJ eval92	-	81 hrs
		PR	WSJ si84/si284	WSJ dev93	-	15/81 hrs
vq-APC [119]	LS 360 hrs	PC	WSJ si284[36]	WSJ dev93	✓	81 hrs
		SID	WSJ si284[35]	WSJ si284[35]	✓	65[34] hrs
vq-wav2vec + DiscreteBERT [22]	LS 960 hrs	ASR	LS 100 hrs	LS test-clean, LS test-other	-	10 mins[34], 1[34]/10[34]/100 hrs
speech-XLNet [615]	LS 960 hrs + WSJ si284 + TED2	PR	TIMIT[30]	TIMIT[30]	✓	4 hrs
		ASR	WSJ si284	WSJ eval92	-	7[34]/14[34]/30[34]/81 hrs
MPC [309, 311]	Didi Callcenter, Didi Dictation, Open Mandarin	ASR-zh	HKUST[30]	HKUST[30]	✓	168 hrs
		ASR-zh	AISHELL-1[30]	AISHELL-1[30]	✓	178 hrs
	SWB, Fisher 1k, LS 960 hrs	ASR	SWB	Hub5'00	-	260 hrs
MPE [414]	WSJ si284, LS 960 hrs	ASR	WSJ si284	WSJ eval92	-	25[34]/40[34]/81 hrs
		ASR	LS 100/360/960 hrs	LS test-clean	-	100/360/960 hrs
ConvDMM [337]	LS 50[547]/360/ 960 hrs	PC/PR	WSJ si284	WSJ eval92	✓	5[34]/50[34]/100[34] mins, 4[34]/8[34]/40[34] hrs
wav2vec 2.0 [26]	LS 960 hrs, LL 60k hrs	ASR	LS 960 hrs	LS test-clean, LS test-other	-	10 mins[34], 1[34]/10[34]/100/960 hrs
		PR	TIMIT[30]	TIMIT[30]	✓	4 hrs
NPC [408]	LS 360 hrs	PC	WSJ si284[36]	WSJ dev93	✓	81 hrs
		SID	WSJ si284[35]	WSJ si284[35]	✓	65[34] hrs
DeCoAR 2.0 [405]	LS 960 hrs	ASR	LS 100 hrs	LS test-clean, LS test-other	-	1[34]/10[34]/100 hrs
TERA [410]	LS 100/360/ 960 hrs	PC	LS 100 hrs[29]	LS 100 hrs[29]	-	80[34] hrs
		SID	LS 100 hrs[29]	LS 100 hrs[29]	-	80[34] hrs
		PR	TIMIT[30]	TIMIT[30]	✓	4 hrs
		ASR	LS 100 hrs	LS test-clean	-	100 hrs
HuBERT [275]	LS 960 hrs, LL 60k hrs	ASR	LS 960 hrs	LS test-clean LS test-other	-	10 mins[34], 1[34] /10[34]/100/960 hrs
wav2vec-c [570]	Alexa-10k	ASR	Alexa-eval	Alexa-eval	✓	1k hrs
UniSpeech-SAT [97]	LL 60k hrs + GigaSpeech-10k + VP-24k	Multi	SUPERB	SUPERB	✓	See SUPERB [723] paper for details
WavLM [98]	LL 60k hrs + GigaSpeech-10k + VP-24k	Multi	SUPERB	SUPERB	✓	See [723] for details
XLS-R [17]	VP-400k + MLS + CV-dataset-7k + VL + BBL	ASR	VP, MLS, CV-dataset, BBL, LS	VP, MLS, CV-dataset, BBL, LS	-	See [17] for details
		SID	VoxCeleb1	VoxCeleb1	✓	
		ST	CoVoST-2	CoVoST-2	✓	
		LID	VL	VL	-	

A.4.4   EXPERIMENT SETTINGS FOR EVALUATING SSL TECHNIQUES

A common way to benchmark SSL techniques and show their efficacy is to fine-tune a pre-trained SSL model for a supervised downstream task. Depending on the corpora used in pre-training and fine-tuning, techniques can be benchmarked in terms of their capability to transfer knowledge across datasets (i.e., using pre-training corpora that differ from the fine-tuning ones), their benefit when training with limited labeled examples (i.e., sampling a subset of labeled examples for fine-tuning), or their improvement over a fully supervised baseline (i.e., using the entire training split of downstream datasets for fine-tuning). Tables A.3 and A.4 summarize experiment settings used in the SSL literature, including the pre-training corpora, downstream tasks and datasets, and the amount of fine-tuning labels used, which indicates the targeted benchmarking scenario as discussed above. Note that there are a variety of ways to fine-tune pre-trained networks (e.g., fine-tune the entire network, freeze certain layers during fine-tuning, and add various architectures of prediction layers to pre-trained networks). We here omit descriptions of these choices; readers can consult the original publications for details.

As observed in tables A.3 and A.4, LS and WSJ are the most commonly used pre-training corpora. At the same time, we observe a growing industry investment in pre-training with larger datasets, e.g., CPC-8k (8k hours) for Bidir-CPC [331], LL (60k hours) for CPC modified [558], wav2vec 2.0 [26], and HuBERT [275], Alexa internal datasets (10k hours) for wav2vec-c [570], Didi internal datasets (10k hours) for MPC [309, 311], the combination of Gigaspeech, VP-24k, and LL (94k hours in total) for UniSpeech-SAT [97] and WavLM [98], and the combination of VP-400K, MLS, CV-dataset, VL and BBL (436k hours in total) for XLS-R [17]. We expect this trend to continue with the growth in available computing power. Most studies focus on learning representations for English, whereas Chinese [309, 311] and multilingualism [17, 331] are also gaining attention. Compared to pre-training, datasets used for fine-tuning are more diverse and cover downstream tasks as varied as ASR, PR, PC, SID, AED, Sentiment, ST, and LID. For benchmarking training scenarios covering full supervision as well as limited resources, the amount of labeled examples used for fine-tuning also varies from several minutes up to 1k hours. Recent benchmarks such as SUPERB [723] that consolidate multiple downstream tasks have gained attention for evaluating SSL

(SSA) [697], TED3, and SWB. SSA is a growing annotated corpus of English speech with various accents. Among the papers studied in this review, SSA is used in [331] only for pre-training, and only 1 hour of audio is utilized. Thus, we exclude it from our discussion in appendix A.4.

[34] A subset of the official training split is sampled, usually to mimic low-resource learning conditions or to quickly evaluate for training and testing on the same split but disjoint subsets.

[35] Dataset split into training, validation, and test subsets at a ratio of 8:1:1.

[36] Dataset split into training and validation subsets at a ratio of 9:1.

[37] The dataset split into training and test subsets at a ratio of 9:1.

methodologies [97, 98]. The goal of such benchmarks is to provide a holistic evaluation of the performance of learned representations; we discuss these in detail in appendix A.4.5. With the increasing popularity of SSL research, we expect future experiment settings to proliferate and cover more languages, downstream tasks, and pre-training/fine-tuning datasets.

A.4.5   Benchmark results and discussion

Given the diversity of datasets and downstream tasks used to evaluate SSL techniques in the literature, it is infeasible to discuss all experiment settings in this survey. Hence, due to their wide adoption for experiments conducted by studies in both SSL and the speech community in general, we focus first on ASR on the LS dataset to understand the efficacy of SSL. We examine SSL techniques which report ASR results on the LS *test-clean* split, and summarize the published WER in figure A.3. The ASR models were obtained first by using unlabeled speech to pre-train a model with each SSL technique. The model was then fine-tuned on labeled data by utilizing a supervised training objective. Respectively, 960, 100, 10, 1 hour(s), and 10 minutes of labeled LS training data were used for fine-tuning, as indicated in different panels of figure A.3 (see the caption of figure A.3 for more details). Semi-supervised methods such as self-training, where a model is first trained on labeled data to annotate unlabeled speech, and then subsequently trained on combined golden and self-annotated label-speech pairs, are gaining popularity in the speech community and have yielded competitive results. For comparison, we also show performance from such methods (iterative pseudo labeling (IPL) [718], slimIPL [401], noisy student [508]), as well as the current state of the art—conformer XXL + noisy student [739]—which augments SSL with various advanced techniques including self-training. Furthermore, we illustrate in the figure the performance of a baseline system [723] based on log Mel filter bank (fBANK), which is one of the most commonly used features designed by domain experts. As observed in the figure, most SSL techniques outperform fBANK features, and with the growing investment in model size, better performance is achieved. The largest ones, such as wav2vec 2.0-L and HuBERT-L/XL, yield competitive results when the entire 960-hour of labeled data is used in training/fine-tuning. The benefit of SSL, especially models with more parameters like wav2vec 2.0 and HuBERT, becomes more evident when the labeling resources become scarce. Compared to popular semi-supervised methods such as IPL, slimIPL, and noisy student using 100 hours of labels, wav2vec 2.0 and HuBERT achieve lower or competitive WERs with 1 hour or even 10 minutes of labeled examples. The results are highly favorable for low-resource use cases, for instance when expanding systems to new domains or languages for which large amounts of unlabeled audio are available, since collecting labels for new
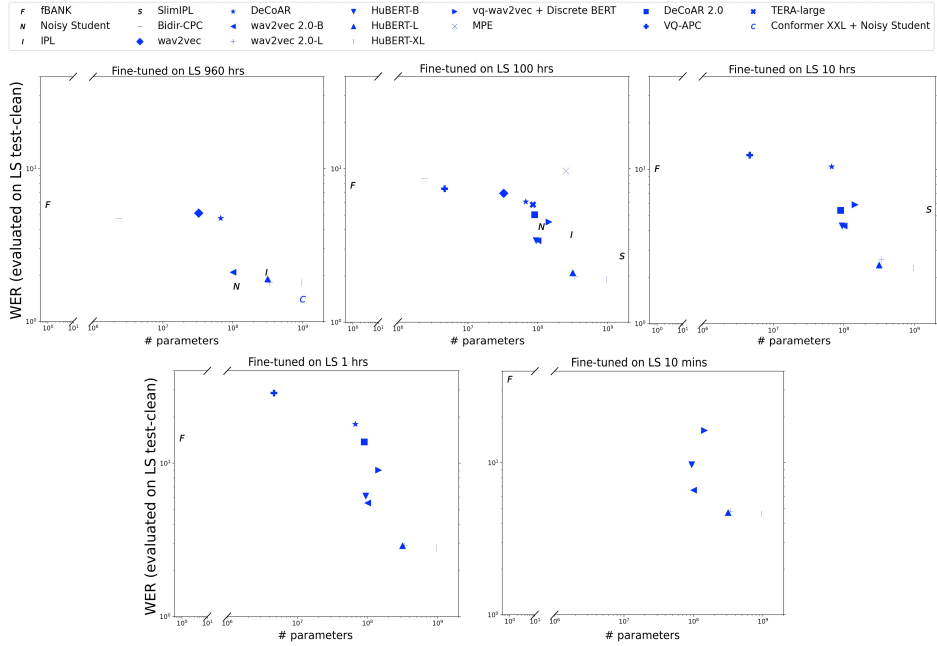
Figure A.3: SSL performance on ASR WER (vertical axis) evaluated with LS *test-clean* split. Techniques are sorted based on the number of model parameters along the horizontal axis. Markers in blue correspond to models initialized with various SSL techniques and then fine-tuned using 960, 100, 10, 1 hour(s), and 10 minutes respectively. The 960-hour training set is the aggregation of *train-clean-100*, *train-clean-360*, and *train-other-500* splits. The 100-, 10-, 1-hour, and 10-minute sets leverage *train-clean-100* or its sampling, except for Bidir-CPC, which samples 10% of the training examples from the entire 960-hour corpus. For simplicity, several SSL techniques are appended with suffixes *B*, *L*, *XL*, or *XXL* indicating the *Base*, *Large*, *X-Large*, or *XX-Large* variants specified in the original publication. We also compare with baselines including the log Mel filter bank (fBANK) and semi-supervised, self-training approaches (iterative pseudo labeling (IPL) [718], slimIPL [401], noisy student [508]). These approaches are visualized in black. Also, note that the current state of the art—conformer XXL + noisy student [739]—is a combination of self-training and SSL techniques. Given the diversity of the listed methods in experiment settings (e.g., pre-training corpora and objectives, whether a language model is used in decoding, whether model parameters are frozen in fine-tuning), readers should be careful that the superiority of methods cannot be decided only based on lower WER numbers.

Table A.5: Tasks where the state of the art is models with SSL pre-training.

Tasks	Dataset	non-SSL	SSL
ASR (WER ↓)	LS test-clean/other	2.1/4.0 [718]	1.4/2.6 [739]
IC (Acc. ↑)	FSC	98.8 [427]	99.3[97]
SID (Acc. ↑)	VoxCeleb1	94.8 [232]	95.5 [98]
ASV (EER ↓)	VoxCeleb1	3.1 [231]	2.4 [692]
QbE (MTWV ↑)	QUESST (EN)	10.6 [561]	11.2[97]

conditions is often prohibitively slow or costly.

In addition to the ASR task, where the current state of the art is achieved by a method combining SSL pre-training and self-training techniques [739], SSL models are competitive in other tasks, including IC, SID, ASV, and QbE. We summarize the performance of these models and previous non-SSL methods in table A.5. The results suggest that the benefit of SSL is generalizable among tasks that require encoding information such as content, speaker, and semantics. As SSL research gains more attention, we expect that SSL pre-trained models will achieve state-of-the-art results on an increasing number of tasks.

Despite the obvious trend of increasing performance as more parameters and SSL pre-training data are being used, numbers in figure A.3 and table A.5 are less comparable than might be expected. The task performance is obtained from the original papers and is often achieved with different downstream fine-tuning recipes, including various language models (used in the ASR system), prediction heads (networks added to SSL for downstream inference), or choices between fine-tuning the whole networks or freezing the SSL encoders. For example, in the ASR task, HuBERT-L and wav2vec 2.0-L leverage Transformer as their language model, while a 4-gram language model trained on LS is used in DeCoAR 2.0. The lack of common and established mechanisms to evaluate SSL techniques in downstream applications makes it difficult to compare techniques fairly and understand their capabilities. To address this challenge, there are increasing efforts to establish benchmarks with shared downstream tasks, datasets, and downstream recipes. Such efforts include SUPERB [723], LeBenchmark [179], ZeroSpeech [166], HEAR [653], NOSS [604], and HARES [686].

SUPERB [723] is a benchmarking platform that allows the SSL community to train, evaluate, and compare speech representations on diverse downstream speech processing tasks, from acoustic and speaker identity to paralinguistic and semantics. SUPERB consolidates downstream recipes to focus on common and straightforward settings (e.g., prediction head architectures, language models, hyperparameter spaces) to facilitate generalizable and reproducible benchmarking of SSL techniques. SUPERB also encourages researchers to innovate for effi-

cient use of model parameters and computation resources to democratize SSL beyond race among Big Tech. LeBenchmark [179] shares a vision similar to SUPERB and provides a reproducible framework for assessing SSL in French with ASR, spoken language understanding, speech translation, and emotion recognition. ZeroSpeech [166] (described in more detail in appendix A.6.2) challenges the scientific community to build speech and language understanding systems using zero expert resources for millions of users of "low-resource" languages. SSL techniques are also benchmarked with the ZeroSpeech challenge [643, 661]. Apart from the speech community, researchers have also established HEAR (holistic evaluation of audio representations) [653], NOSS (non-semantic speech benchmark) [604], and HARES (holistic audio representation evaluation suite) [686] to benchmark audio representations. These efforts promote the creation of an audio embedding that is as holistic as the human ear in interpreting speech, environmental sound, and music. Given the significant need to understand and compare SSL techniques fairly and comprehensively, we expect SSL benchmarking to remain an active research area.

A.5   ANALYSIS OF SELF-SUPERVISED REPRESENTATIONS

The previous sections have shown how self-supervised learning can result in powerful representations that provide a robust starting point for several downstream tasks. It is natural to ask if we can gain an even deeper understanding of the nature of these representations, in order to further optimize them or apply them to different problems. What is the information encoded in these representations? How robust are they to distributional shifts, and how dependent are they on the size of the training data? Do they generalize across languages? What are the key ingredients for training powerful representations: input data, network architecture, training criterion, or all three? Can we predict their performance on downstream tasks from their training behavior? This section tries to answer these questions by summarizing several studies that analyze self-supervised representations.

A.5.1   INFORMATION CONTENT

In [513] wav2vec 2.0 representations were analyzed with respect to their acoustic-linguistic information content at different network layers. Three different mechanisms were used for this purpose. The first of these is canonical correlation analysis (CCA), which computes similarity scores between two continuous vectors based on the maximum correlation of their linear projections. These can be used to judge the similarity of embeddings at different layers with each other, with standard acoustic representations such as Mel filter bank features, or word

embeddings derived from text. The second method clusters continuous representation vectors and computes the discrete mutual information between cluster IDs and phone or word labels. The third method involves probing tasks: representation vectors extracted from the network are used to perform simple downstream tasks, in particular determining whether two acoustic segments correspond to the same word, and a standard benchmark of 11 word similarity tasks [180]. These are mostly used to gauge the amount of lexical information present in the embeddings. Using this battery of tests the authors compared pre-trained models of varying sizes as well as models fine-tuned for ASR. They found that pre-trained models show an autoencoder-like behavior, with early layers showing strong similarity with input features, intermediate layers diverging more, and final layers reverting to higher similarity with input features and early layers. Generally, the earlier layers in wav2vec 2.0 models encode acoustic information. The next set of layers encodes phonetic class information, followed by word meaning information, before reverting to encoding phonetic/acoustic information. Thus, extracting representations from the last layers for tasks that require phonetic or word-related information may not be the best strategy. Indeed, the authors of [23] show that a phone classifier trained on each of the 24 frozen layers of a wav2vec 2.0 model showed the lowest phone error rates for layers 10–21 and higher error rates for the other layers. [513] further show that fine-tuning the pre-trained model with a character-level CTC training criterion changes the behavior of the last layers (especially the final two layers), breaking the autoencoder-like behavior and focusing the information encoded in the last layers on orthographic-phonetic and word information.

The peaking of class-relevant information in intermediate layers seems to be common across different self-supervised learners and different modalities. In an analysis of text-based Transformers trained with a masked language model criterion [670] observed a similar compression plus reconstruction pattern. Interestingly, similar network behavior was also recently described for self-supervised learners in computer vision: using a contrastive self-supervised learner (SimCLR) that optimizes for augmentation invariance, [220] show that it is the intermediate representations that most closely approximate information learned in a supervised way, i.e., they provide more class information than the representations from final layers. This is similar to the findings described above for wav2vec 2.0 without fine-tuning, where intermediate layers provide more information about phone and word classes.

Self-supervised representations may encode other information besides phonetic classes or words, for example, channel, language, speaker, and sentiment information. It is shown that the per-utterance mean of CPC features captures speaker information to a large extent[660]. Location of information pertaining to speakers vs. language classes was analyzed in [407] for a 12-layer BERTphone

model. This model combines a self-supervised masked reconstruction loss with a phone-based CTC loss to produce representations for speaker recognition and language identification. By analyzing the weights of a linear combination of layer representations for these two downstream tasks, it was shown that language recognition draws on representation from higher layers (peaking at layer 10) whereas speaker recognition benefited from layers at positions 6, 9, and 12. This may indicate that language recognition relies more on higher-level phonetic information whereas speaker recognition uses a combination of acoustic and phonetic information. In a recent study [98] the same technique was used to identify layer contributions for the downstream SUPERB benchmark tasks in the WavLM model. For a smaller model (95M parameters) it was again confirmed that lower layers encode speaker-related information necessary for speaker diarization and verification whereas higher layers encode phonetic and semantic information. Another study [678] used explicit self-supervised loss at the intermediate layers rather than just the output layer of a HuBERT model in order to enforce better learning of phonetic information. The resulting model was indeed better at downstream tasks requiring information about phonetic content, such as phone recognition, ASR, and keyword spotting, but worse at speaker-related tasks like speaker diarization and verification.

Most self-supervised learning approaches rely on a Transformer architecture for the representation model. In [722] the attention patterns in generatively trained Transformer representation models were analyzed. Self-attention heads were grouped into three categories: diagonal, vertical, and global. It was found that the diagonal head focuses on neighbors and is highly correlated with phoneme boundaries, whereas the vertical head focuses on specific phonemes in the utterance. Global heads were found to be redundant as removing them resulted in faster inference time and higher performance.

A.5.2 TRAINING CRITERION

In [113], representations based on different training criteria (masked predictive coding, contrastive predictive coding, and autoregressive predictive coding) were compared and analyzed with respect to the correlation between their training loss and performance on both phone discrimination and speaker classification probing tasks. It was observed that the autoregressive predictive coding loss showed the strongest correlation with downstream performance on both tasks; however, models were not further analyzed internally. An evaluation of the similarity of representations trained according to the three criteria above (but with different architectures and directionality of contextual information) also showed that it is the training criterion that most influences the information encoded in the representations, not the architecture of the learner or the directionality of the

input.

A similar insight was obtained in [743], which compared VQ-VAE and vq-wav2vec with respect to their ability to discover phonetic units. The VQ-VAE model extracts continuous features from the audio signal; a quantizer then maps them into a discrete space, and a decoder is trained to reconstruct the original audio conditioned on the latent discrete representation and the past acoustic observations. By contrast, vq-wav2vec predicts future latent discrete representations based on contextualized embeddings of past discrete representations, in a CPC-style way. The models were evaluated according to their ability to discover phonetic units (as measured by phone recognition error rate on TIMIT, and the ZeroSpeech ABX task (see appendix A.6 for more details)), and it was found that the predictive vq-wav2vec model fared better than the autoencoder-like VQ-VAE model, most likely due to its superior ability to model temporal dynamics.

A.5.3 EFFECTS OF DATA AND MODEL SIZE

How does the performance of self-supervised models change in relation to the amount of training data, and in relation to the size (number of parameters) of the model? Several studies have demonstrated better downstream performance when using larger datasets [98, 331, 558]. For example, [331] compared representations learned by a bidirectional CPC model from the standard 960 hour LS corpus and a corpus of 8,000 hours of diverse speech from multiple sources.[33] Not surprisingly, an ASR model trained on top of these representations performed better when representations were learned from the larger dataset. Although the precise relationship between data size and performance has not been quantified, we can assume that it follows a law of diminishing returns (or power law), similar to observations for most data-intensive machine learning tasks. In addition to the size of the dataset, the diversity of the data also seems to play a role, although this was not quantified in this study. However, recent experiments with larger and more diverse data collections [98] confirm this assumption, as do explicit investigations of domain shift robustness (see appendix A.5.4 below).

The relation between model sizes and downstream performances have also been investigated [533, 667]. Using the Mockingjay model [411], the authors in [533] attempt to establish a relationship between model size and self-supervised L_1 loss and demonstrate that it approximately follows a power law. Model size and accuracy on downstream phone classification and speaker recognition tasks are positively correlated but do not exactly follow a power law; rather, the accuracy saturates as models increase in size, possibly due to the lack of a corresponding expansion in training data size.

A.5.4   ROBUSTNESS AND TRANSFERABILITY

It is well known that traditional speech features like MFCCs lack robustness against environmental effects such as additive noise, reverberation, accents, etc., that cause differences in the distributions of speech features. Do pre-trained representations offer greater robustness against distributional shifts? One study [331] compared pre-trained representations from a CPC model against MFCCs and found pre-trained representations to be more robust to mismatches between training and test data. The training data consisted of clean, read speech (LS) whereas test data consisted of the Switchboard corpus and TED talks. The distributional shifts here may stem from both the acoustics (microphone, room reverberation), lexical effects related to topic and style, and differences in speaker characteristics such as accent. Similar problems were also investigated using HuBERT and wav2vec 2.0 models in [90]. In [276] domain effects were studied in greater detail using datasets from six different domains. In particular, the authors focused on the usefulness of adding out-of-domain data to pre-training. The general conclusions are that pre-training on more and diverse domains is preferable: models pre-trained on more domains performed better than those pre-trained on fewer when tested on held-out domains, regardless of which additional labeled data was used for fine-tuning. Adding in-domain unlabeled data—if available—to pre-training improves performance robustly; however, even out-of-domain unlabeled data is helpful and closes 66–73% of the performance gap between the ideal setting of in-domain labeled data and a competitive supervised out-of-domain model.

In [558] the effectiveness of CPC-trained representations for phone discrimination tasks was compared across several languages. It was found that representations pre-trained only on English successfully enabled phone discrimination in 10 other languages, rivaling supervised methods in accuracy in low-data regimes (1h of labeled data per language). Thus, self-supervised pre-training enables the model to learn contextualized speech features that generalize across different languages. In [131], a wav2vec 2.0 model was trained on data from multiple different languages and different corpora (Babel, Common Voice, and multilingual LS) jointly, followed by fine-tuning for each individual language. The largest model covers 53 languages in total and consists of 56,000 hours of speech. Compared to monolingual pre-training, even smaller models trained on only ten languages improve performance substantially on a downstream character-based ASR task. Low-resource languages with little labeled data improve the most under this training regime. Multilingual representations also resulted in competitive performance (lower character error rate than monolingual representations) for languages not present in the training dataset, again showing that unsupervised pre-trained representations can learn generic features of the speech signal that generalize across different languages. The study also found that sharing

data from closely related languages is more beneficial than combining distant languages. An analysis of language clusters in the shared discrete latent representation space revealed that similar languages do indeed show a higher degree of sharing of discrete tokens. Finally, one might ask whether the interpretation of representations extracted from different layers of a self-supervised models also generalizes to the multilingual setting. Experiments in [23] on phone recognition in eight languages based on the different layers of the multilingual wav2vec 2.0 XLSR-53 model indicate that this is indeed the case: phone error rates showed the same pattern as in the monolingual (English) scenario, with lower phone error rates for middle layers as opposed to earlier/later layers.

A.6 From representation learning to zero resources

In the SSL framework, speech representations can be learned and used in various downstream tasks to achieve competitive, robust, and transferable performance, as shown in appendices A.4 to A.5. However, labeled data is still required. For example, in ASR, utterances and their manual transcriptions are needed to learn downstream models or fine-tune representation models. Can a model learn without any labeled data? In appendix A.6.1, we show how to learn ASR models without any paired audio and text and how SSL improves the framework. In addition, many languages have no writing system. In appendix A.6.2, the SSL representation is further used in scenarios where text data is unavailable.

A.6.1 Unpaired text and audio

Unsupervised ASR If only unpaired speech and text are available, that is, the text is not a manual transcription of speech, can the machine learn how to transcribe speech into text? This scenario is called *unsupervised ASR*, and the framework is as below. Given a set of unlabeled utterances $\mathcal{S} = \{S_1, S_2, ..., S_N\}$ and a set of sentences $\mathcal{Y} = \{Y_1, Y_2, ..., Y_M\}$,[38] a mapping function F, which can take an utterance S as input and generate its transcription, is learned from data. Table A.6 summarizes recent work on unsupervised ASR, including the speech representation used, the algorithm used to learn the mapping without supervision, and the results. Below, we will discuss these methods in more detail.

Adversarial training [11, 211, 225] is one common way to learn such a mapping function. The framework includes a discriminator and a generator. The mapping function F plays the role of the generator, which takes speech utterances as input and outputs text. The discriminator learns to distinguish real text from the generated output; the generator learns to "fool" the discriminator. The

---

[38]Note that the speech and text are not paired, that is, Y_i is not the transcription of S_i.

Table A.6: Unsupervised ASR. TIMIT numbers are phoneme error rates (PER), while the numbers for LibriSpeech are word error rates (WER). SWC = spoken word classifier, ST = speech translation. All speech and text are in English if not specified. The references in the table are sorted according to the date of publication.

Reference	Speech representation	Speech segmentation	Token	Mapping approach	Refinement	Results
[416]	Audio word2vec [684]	Oracle	Phoneme	Adversarial Training [225]	-	TIMIT (PER): 63.6%
[121]	Speech2vec [117]	BES-GMM [323]	Word2Vec	Adversarial Training [132]	Self-training	SWC (Acc.): 10.9%
[120]	Speech2vec (English)	Oracle	Word2Vec (French)	VecMap [13]	LM rescore, sequence DAE	ST (BLEU): 10.8%
[726]	MFCC	GAS [683]	Phoneme	Empirical-ODM [422]	Self-training	TIMIT (PER): 41.6%
[96]	MFCC	GAS	Phoneme	Adversarial Training [225]	Self-training	TIMIT (PER): 33.1%
[23]	Wav2vec 2.0 [26]	k-means	Phoneme	Adversarial Training [225]	Self-training	TIMIT (PER): 18.6%, LibriSpeech (WER): 5.9%
[349]	Universal Phone Recognizer	-	Grapheme	Deciphering [550]	Self-training	GlobalPhone: 32.5% to just 1.9% worse than supervised models
[409]	Wav2vec 2.0 [26]	-	Phoneme	Adversarial Training [225]	Self-training	LibriSpeech (WER): 6.3%
[409]	Wav2vec 2.0 [26]	-	Grapheme	Adversarial Training [225]	Self-training	LJSpeech (WER): 64.0%

generator and the discriminator are trained in an iterative, interleaved way. After the training, the generator serves as the speech recognition model. There is a large amount of work using gradient penalty in the objective of training discriminators [23, 96, 409, 416], which is inspired by Improved Wasserstein Generative Adversarial Network (WGAN) [225]. Other ways to map speech and text include segmental empirical output distribution matching (segmental empirical-ODM) [726] and the deciphering algorithm [349].

Success in unsupervised neural machine translation (MT) [14, 132, 371] has inspired innovative exploration of various unsupervised ASR algorithms. If learning a translation model from unaligned sentences in two languages is possible, considering speech and text as two different languages, learning the mapping relationship from speech space to text space without an alignment should likewise be possible. However, there are differences between unsupervised MT and unsupervised ASR. In unsupervised MT, most discrete source tokens can be mapped to specific target tokens representing the same meaning. However, because speech has segmental structures, in unsupervised ASR, each text token maps to a segment of consecutive acoustic features of variable length in an utterance. The generator is supposed to learn the segmental structure of an utterance because information like token boundaries is not directly available. This makes unsupervised ASR more challenging than unsupervised MT.

For unsupervised ASR to be feasible, the common idea is to make the speech and text units close to each other. For the text side, word sequences can be transformed into phoneme sequences if a lexicon is available. On the other hand, we must first convert the speech signal into something close to phonemes. To achieve that, most studies on unsupervised ASR use a phoneme segmentation module before the generator to segment utterances into phoneme-level segments [96, 416, 726]. A representation vector or a token then represents each phoneme-level segment. It is easier for the generator to map each segment-level representation or token to the correct phoneme when the representation or token is highly correlated to the phonemes. Wav2vec-U [23] selects the input feature from different layers of wave2vec 2.0 [26]. The selection criterion is based on analysis of the phonetic information in each layer. If a universal phone recognizer trained from a diverse set of languages is available, it is another way to transcribe speech into phone-level tokens [349]. Another series of work is to transform a word into a word embedding. [120, 121] use adversarial training to map the word-level speech embedding space [117] to the word embedding space and achieve promising performance on spoken word classification, speech translation, and spoken word retrieval. Table A.6 summarizes the various ways to segment speech and represent speech and text in each reference.

As shown in table A.6, most studies use *self-training* to refine the models. In self-training, the generator serves as the first-version phoneme recognition

model. Inputting unpaired speech to the generator generates the corresponding "pseudo transcription". We then view the speech utterances and their pseudo transcriptions as paired data which we use to train a model in a supervised manner. Although the pseudo transcriptions have more errors than oracle transcriptions, experiments show that training models on pseudo transcriptions still significantly boosts performance compared to the first-version model.

Wav2vec-U [23] achieved state-of-the-art results at the time, which suggests that representation learning is essential for the success of unsupervised ASR. It achieved an 11.3% phoneme error rate on the TIMIT benchmark. On the LS benchmark, wav2vec-U achieved a 5.9% WER on *test-other*, rivaling some of the best published systems trained on 960 hours of labeled data from only two years earlier. And wav2vec-U 2.0 [409] further removes the requirement of the segmentation stage, so the unsupervised ASR model can be learned in an end-to-end style. The robustness of wav2vec-U was further analyzed with respect to domain-mismatch scenarios in which the domains of unpaired speech and text were different [403]. Experimental results showed that domain mismatch leads to inferior performance, but a representation model pre-trained on the targeted speech domain extracts better representations and reduces this drop in performance.

ASR-TTS   Here we describe an alternative approach by which to train an ASR and text-to-speech (TTS) system based on unpaired text and audio. The ASR-TTS framework, which combines the ASR and TTS systems in a cascaded manner, can be regarded as an autoencoder, where the encoder f corresponds to the ASR module and the decoder g corresponds to the TTS module. In this framework, we consider the intermediate ASR output as a latent representation; the framework as a whole can be regarded as a variant of self-supervised learning.[39]

The ASR-TTS framework can jointly optimize both ASR and TTS without using paired data [271, 641, 680]. A speech chain [641, 642] is one successful way to utilize audio-only and text-only data to train both end-to-end ASR/TTS models. This approach first prepares pre-trained ASR model $f_{asr}(X)$ with acoustic input X and pre-trained TTS model $g_{tts}(Y)$ with text input Y. By following the TTS system with an ASR system, we generate new acoustic feature sequence $\hat{X}$, which must be close to the original input X. Thus, we design a loss function $\mathcal{L}_{asr \to tts}(X, \hat{X})$, where $\hat{X}$ is generated by

$$\hat{X} = g_{tts}(f_{asr}(X)) \ . \tag{A.36}$$

---

[39]However, to make this complicated system work, we often require that data is paired. Therefore, in practice, ASR-TTS and other methods described in this section are categorized as semi-supervised learning.

Thus, we train the ASR model (or both ASR and TTS models) using only the acoustic input by minimizing $\mathcal{L}_{asr \to tts}$. Note that this approach does not require the supervised text data Y. As an analogy to the generative approach in appendix A.3.2, the intermediate ASR output $\hat{Y}$ can be regarded as the latent representation Z.

The other cycle with the text-only data Y is also accomplished by the concatenated TTS-ASR systems:

$$\hat{Y} = f_{asr}(g_{tts}(Y)) \ . \tag{A.37}$$

Similarly, this approach does not require the supervised audio data X, and the intermediate TTS output $\hat{X}$ can be regarded as the latent representation Z. Although this approach initially freezes either the ASR or TTS model, extensions of this study [34, 271, 640] implement the joint training of both ASR and TTS parameters using REINFORCE [701] and straight-through estimators.

An emerging technique uses a well-trained TTS system to generate speech and text data from text-only data. This technique is a sub-problem of the TTS-ASR approach formulated in (A.37) in which we fix the TTS system part and estimate only the ASR parameters. For example, a huge amount of text resources can be obtained from the web and document archives without corresponding audio data. The typical use case scenario of such a text resource for ASR is through the language model. We combine the ASR and language model via a noisy channel model [306], a weighted finite state transducer [463], or shallow fusion [109, 224]. However, the progress of TTS systems boosted by deep learning [496, 597] has inspired another interesting and straightforward research direction: artificially creating paired text and audio data $\{\hat{X}, Y\}$ with only text data Y by generating the corresponding audio data $\hat{X}$ with TTS. The most straightforward approach is to simply use multi-speaker TTS to generate the waveform with various acoustic variations [286, 372, 399, 564, 654]. The other approaches are based on the generation of high-level (more linguistic) features instead of generating the waveform, e.g., encoder features [248] and phoneme features [445, 553]. This approach is similar to the back-translation technique developed in neural machine translation [591]. One benefit of the above data generation approaches is that it can be used to feed unseen word or context phrases to end-to-end ASR.

A.6.2   No text or lexicon

Zero-resource speech technologies and challenges   Zero-resource speech technologies, which seek to discover linguistic concepts from audio only (no text nor lexicon), are one of the most active applications of unsupervised/self-supervised speech processing. Zero-resource speech technologies were initially studied for acoustic and linguistic unit discovery from speech data without linguistic re-

sources, e.g., transcriptions and other annotations [297]. This study was motivated by unsupervised query-by-example, applications of nonparametric Bayesian machine learning to speech processing, and low-resource speech recognition, and was also inspired by the learning process of infants. The goal of this type of work is to build spoken dialog systems in a zero-resource setup for any language. To encourage zero-resource research, zero-resource speech challenges have been organized since 2015.

In this section, we describe the research directions of zero-resource speech technologies by following the series of zero-resource speech challenges.

- Zero Resource Speech Challenge 2015 [667] mainly focused on building an acoustic model without using any linguistic annotations based on subword unit modeling and spoken term discovery tracks. For the subword unit modeling track, the ABX score for the within- and across-speaker tasks was used as an evaluation metric. The spoken term discovery track used the normalized edit distance and coverage scores in addition to the precision, recall, and F1-scores for types, tokens, and boundaries. Both tracks were based on the English and Xitsonga languages.

- The Zero Resource Speech Challenge 2017 [165] focused on unseen language and speaker aspects from the previous challenge. For example, to demonstrate the robustness against unseen languages, the systems were developed with English, French, and Mandarin and tested on two "surprise" languages: German and Wolof. Similarly, robustness against unseen speakers was demonstrated by varying the amount of speech available for each speaker.

- The Zero Resource Speech Challenge 2019 [163] extended a goal of previous challenges by synthesizing speech without text or phonetic labels but with acoustic units obtained using zero-resource techniques. The evaluation metrics were also extended to subjectively evaluate the quality of synthesized speech, including its intelligibility, naturalness, and speaker similarity.

- The Zero Resource Speech Challenge 2020 [166] was based on two tracks, revisiting previous challenges with different evaluation metrics. The first task revisited the 2019 challenge with low bit-rate subword representations that optimize the quality of speech synthesis. The second task revisited the 2017 challenge by focusing on the discovery of word-like units from unsegmented raw speech.

- The Zero Resource Speech Challenge 2021 [486], the latest challenge, expanded the scope to include language modeling tasks. In addition to phoneme-

level ABX, the challenge includes lexical, semantic, and syntactic evalua-
tion metrics computed via a language model of pseudo-acoustic labels.

These challenges have facilitated the tracking of technical trends in zero-resource
speech technologies. For example, research directions thereof have expanded to
various speech processing components to cover the entire spoken dialogue sys-
tems. To keep up with this expansion, the challenge has continued to develop ap-
propriate evaluation metrics for zero-resource scenarios. Following the success
of representation learning, baseline and challenge techniques have shifted from
purely generative models [251, 495], deep autoencoders [110, 644], and incor-
poration of neural-network-based TTS/VC techniques [661] to self-supervised
learning [441]. The latest challenge included the visual modality, continuing the
expansion to include more aspects of human interaction.

Textless NLP   Textless NLP is a new research direction that leverages the progress
mentioned above in self-supervised speech representation learning to model lan-
guage directly from audio, bypassing the need for text or labels [334, 355, 487,
529]. Not only does this open the gate for language and dialect modeling without
orthographic rules, but it also offers the opportunity to model other non-lexical
information about how speech is delivered, e.g., speaker identity, emotion, hes-
itation, interruptions. The generative spoken language model (GSLM) [487] uti-
lizes discrete representations from wav2vec 2.0, HuBERT, and CPC algorithms as
inputs to an autoregressive language model trained by using the cross-entropy
function to maximize the probability of predicting the next discrete speech to-
ken. A synthesis module follows the language model to produce speech wave-
forms given the generated discrete speech units. The generated spoken contin-
uations compete with supervised generations and synthesis using a character
language model in subjective human evaluations. The model completes incom-
plete words (pow[...] → POWER) and continues using words in the same general
mood (dark → BLACKNESS)[40] and has been extended to model and generate
dialogues [487].[41] Given its flexibility in modeling spoken content, the GSLM
has been further extended to jointly model content and prosody [334]. This
prosodic-GSLM model introduced a multistream causal Transformer, where the
input and output layers use multiple heads to model three channels: discrete
speech units, duration, and quantized pitch. The prosodic-GSLM model jointly
generates novel content and prosody congruently in the expressive style of the
prompt.[42] Going one step further, [355] used a speech emotion conversion frame-
work to modify the perceived emotion of a speech utterance while preserving its

---

[40]https://speechbot.github.io/gslm/
[41]https://speechbot.github.io/dgslm/
[42]https://speechbot.github.io/pgslm/

lexical content and speaker identity. Other studies have extended the idea of text-less language processing or audio discrete representation to applications such as spoken question answering [402], speech separation [601], TTS [247], and speech-to-speech translation [381].

A.7 DISCUSSION AND CONCLUSION

In this overview, we have presented the historical context of self-supervised learning and provided a thorough methodological review of important self-supervised speech representation models. Specifically, we have categorized the approaches into three categories, generative, contrastive and predictive, differing in terms of how the pretext task is defined. We have presented an overview of existing benchmarks and reviewed the efforts towards efficient zero-resource learning. Although the field is progressing rapidly, with new approaches reaching higher levels of performance, a couple of patterns have emerged: (1) The solid performance of Wav2vec 2.0 for speech recognition and many downstream tasks, as well as the public availability of its pre-trained multilingual variants, enabled wide adoption in the community making it a "standard" go-to model. (2) The simplicity and stability of the HuBERT approach, as well as the resemblance of its training procedure to classic frame-level ASR systems, made it an easy choice for research extensions on improving representation quality, speech translation, and textless NLP.

Below we highlight various shortcomings of existing work and future research directions:

- **Using the representation model.** So far, there are two main ways to use representation models: Freeze the representation models and use them as feature extractors, or fine-tune the representation models on downstream tasks. Some efficient methods for leveraging SSL models exist in the NLP community. Adapters [227, 273, 735] are lightweight modules inserted into SSL models, and in downstream tasks, the parameters of SSL models are frozen, and only the adapters are trained. The prompt/instruction learning methods [415] also freeze the SSL parameters and control the output of SSL by adding additional information, which is called *prompt*, in the input. Both adapter-based methods and prompt/instruction learning yield competitive performance compared with fine-tuning in NLP applications, but there is only little related work for speech [89, 637]. In addition, prompt for speech SSL does not achieve comparable performance on sequence generation tasks like phoneme recognition and slot filling, so how to use prompt is still an open question.

- **Increasing the efficiency of the representation model.** As discussed in appendix A.5.3, larger representation models lead to better downstream performance. Despite the success of these large models, they incur high costs in terms of memory and time for pre-training, fine-tuning, and even when used only to extract representations without gradient calculation. This makes them unsuitable for edge devices but also limits the ability to scale these models to very large datasets – and leads to a large energy consumption. Preliminary studies have been conducted on compressing speech representation models through network pruning [362] or knowledge distillation [88]. There has been quite some effort towards more efficient general neural network models via conditional computing [39] and neural network quantization [204] as well as extensive work on improving the specific efficiency of Transformer models, especially with the focus on self-attention [632], but these technology has not been widely used in speech SSL. Because speech is intrinsically represented as sequence, one way to reduce computation is to reduce the length of speech representation sequence but still keep the vital information in speech. But we have not been aware of any publication in this direction when writing this paper. On the other hand, non-streaming architectures in models such as the bidirectional Transformer have hindered the representation model used in streaming scenarios, leading to studies that address these problems [76]. We anticipate research in these directions to continue in the future.

- **Data-efficient approaches.** SOTA representation learning methods require large volumes of unlabeled speech during pre-training, going way beyond what babies need to understand language. Different learning approaches have different data needs, e.g., generative approaches could be more data efficient than contrastive or predictive approaches since they are constrained by more bits of information to reconstruct their inputs. Comprehensive research is needed to study the data efficiency of different approaches.

- **Feature Disentanglement.** Speech SSL models show strengths on a surprisingly wide range of tasks [723], suggesting that representations contain different information. One way to further improve downstream tasks is to disentangle different information from the representation. For example, we can decompose the representation into content embedding and speaker embedding and use content embedding for ASR and speaker embedding for SID. Some work has been in this direction [84, 106, 534].

- **Creating robust models.** As discussed in appendix A.5.4, studies have been conducted on the robustness of representation models [709]. However, the failure modes of SSL models are still poorly understood, and it

remains unclear whether they provide more or less robustness to adversarial attacks than fully supervised models. Due to the importance of this research direction, while writing this paper, there is already some related research about enhancing the robustness of SSL models [276, 282, 682, 746] and identifying their vulnerability to adversarial attack [709].

- **Capturing higher-level semantic information.** Although many representation learning approaches can go beyond low-level phonetic modeling to capture some lexical information [488], they still struggle in higher-level semantic tasks easily captured by word-level counterparts like BERT. One workaround is two-stage training [334, 487]; however, this prevents propagating rich lexical and semantic knowledge modeled in the second stage to benefit the phonetically focused first stage.

- **Using text representation models to improve speech representation.** The amount of content information in speech corpora used to train speech representation models is far less than that of text representation models. Noting that the BERT training corpus exceeds 3 billion words [151], and assuming a typical speaking rate of 120 words per minute, a speech corpus containing the same content as the BERT training data would include 400,000 hours of audio, which exceeds the accumulated training data of all current speech representation models. Therefore, to enable speech representation models to better learn human language, for instance by extracting semantic information from acoustic signals, the use of text models such as BERT and GPT seems key: nevertheless, how to use these to improve speech representation model pre-training remains an open question.

We believe SSL representation models have considerable room to grow. The relationship between representation models and downstream tasks can be compared to the relationship between operating systems and applications. Today, even individuals can build applications with desired functions on a smartphone because the smartphone's operating system handles the complex communication with the hardware and provides a convenient developer interface. Likewise, as SSL representation models learn general knowledge from human speech, it is easy to develop new speech processing applications on this basis. From this viewpoint, speech representation models will play the role of operating systems in speech processing and further facilitate the continued development of speech technology.

SUPPLEMENTARY MATERIAL: HIERARCHICAL VAEs KNOW WHAT THEY DON'T KNOW

B.1 DATASETS

Table B.1 lists the datasets used in the paper. We use the predefined train/test splits for the datasets. For SmallNORB and Omniglot we resize the original grayscale images to 28×28 with ordinary bi-linear interpolation. For each of these datasets, we also create a version where the grayscale is inverted. We do this because, the overall white nature of the images tends to make detecting them as OOD from FashionMNIST artificially easy. The inversion is done via the simple transformation $\mathbf{x}_{\text{inverted}} = 255 - \mathbf{x}_{\text{original}}$ since images are encoded as 8-bit unsigned integers.

Table B.1: Overview of the used datasets.

Dataset	Dimensionality	Examples
FashionMNIST [714]	$28 \times 28 \times 1$	70,000
MNIST [380]	$28 \times 28 \times 1$	70,000
notMNIST [68]	$28 \times 28 \times 1$	547,838
KMNIST [128]	$28 \times 28 \times 1$	70,000
Omniglot [367]	$28 \times 28 \times 1$	32,460
SmallNORB [374]	$28 \times 28 \times 1$	97,200
CIFAR10 [356]	$32 \times 32 \times 3$	60,000
SVHN [480]	$32 \times 32 \times 3$	99,289

B.2 MODEL DETAILS

In table B.2 we specify the hyperparameters used when training our models. We make our source code available at `https://github.com/JakobHavtorn/hvae-oodd`.

B.2.1 HIERARCHICAL VAE

Our Hierarchical VAE (HVAE) model uses bottom-up inference and top-down generative paths as specified in the paper. For grayscale images, the output is

parameterized by a Bernoulli distribution while for natural images we use a Discretized Logistic Mixture [573]. The latent variables are parameterized by stochastic layers that output the mean and log-variance of a diagonal covariance Gaussian. The prior distribution on the top-most latent is a standard Gaussian. For grayscale images, the lowest latent space is parameterized by a convolutional neural network and has dimensions $14 \times 14 \times 8$ interpreted as (height × width × latent dimension). The highest two latent variables are parameterized by dense transformations with 16 and 8 units, respectively. For natural images, the bottom-two latent variables are parameterized by convolutional neural networks and have dimensions $(16 \times 16) \times 128$, $(8 \times 8) \times 64$, respectively for $\mathbf{z}_1, \mathbf{z}_2$. The top-most latent, $\mathbf{z}_3$, is densely connected with dimension 32.

Each stochastic layer is preceded by a deterministic transformation. For both grayscale and natural images, each deterministic transformation consists of three residual blocks of the same type used by Maaløe et al. [434]. The structure of a residual block is:

$$\mathbf{y} = \text{CONV}\left(\text{ACT}\left(\text{CONV}_s\left(\text{ACT}(\mathbf{x})\right)\right)\right) + \mathbf{x} \,, \tag{B.1}$$

where CONV refers to a same-padded convolution and ACT to the activation function. Within a residual block, the first convolution always has stride 1 while the second convolution has stride s. In a deterministic transformation, any non-unit stride is performed in the third residual block. For grayscale images, we stride by 2 in the first and second deterministic transformations but not the third. For natural images, we similarly stride by 2 in the first and second deterministic transformations. For grayscale we use 64 channels while we use 256 for natural images. In both cases, the first deterministic block uses a kernel size of 5 and the latter two a kernel of size 3. We use the ReLU activation function [190, 471].

Since the benefits and drawbacks of using batch normalization [292] in hierarchical VAEs is still the matter of some debate [102, 614, 657] we choose to use weight normalization [574] as in other work [434] and initialize the model using the originally proposed data-dependent initialization. To have the stochastic layers initialize to standard Gaussian distributions (zero mean, unit variance), with this initialization, we select the activation function for the variance as a softplus,

$$\text{SOFTPLUS}(\mathbf{x}) = \frac{1}{\beta} \log\left(1 + \exp(\beta\mathbf{x})\right) \,,$$

with $\beta = \log(2) \approx 0.693$ to output 1 for $\mathbf{x} = 0$.

Training of a HVAE model took approximately two days on a single NVIDIA GTX 1080 Ti graphics card.

B.2.2   BIVA

For the BIVA model [434], we use a specification that is very similar to that of the HVAE above, and to that of the original paper. The model has 10 latent variables the lowest 3 of which are spatial and the rest are densely connected in order to have an architecture similar to the HVAE. The model uses an overall stride of 8, achieved by striding by 2 in the first, fourth and sixth deterministic transformations. From $\mathbf{z}_1$ to $\mathbf{z}_{10}$, the latents have the following dimensions: The lowest three latents are spatial $(16 \times 16) \times 8$, $(16 \times 16) \times 16$ and $(16 \times 16) \times 32$, given as (height $\times$ width) $\times$ dim), while the rest are dense vectors with dimensions of $42, 40, 38, 36, 34, 32, 30$.

Training of a BIVA model took approximately a week on a single NVIDIA GTX 1080 Ti graphics card.

B.3   Analysis of the influence of latent variables on the marginal likelihood

In the paper, we argue that the lowest level latent variables, which have the highest dimensionality, contribute the most to the approximate likelihood. Here, we provide a stringent mathematical argument that generalizes this to the exact marginal likelihood in a model with a deterministic decoder.

B.3.1   Model specification

For an arbitrary hierarchical latent variable model, we have a prior $p(\mathbf{z}_L)$ and a generative mapping $f : \mathbb{R}^d \to \mathbb{R}^D$, such that $\mathbf{x} = f(\mathbf{z}_L)$ and $D > d$. Note that we will assume that f is deterministic, such that we are effectively working with $p(\mathbf{x}|\mathbf{z}) = \delta_{f(\mathbf{z})}(\mathbf{x})$. This is a limiting assumption, but it allows working through the following. For shorthand we will simply write $\mathbf{z} = \mathbf{z}_L$.

Let f have a bottleneck architecture, i.e.

$$f(\mathbf{z}) = f_1(\dots f_{L-1}(f_L(\mathbf{z}))) \,, \tag{B.2}$$

where

$$f_i : \mathbb{R}^{d_i} \to \mathbb{R}^{d_{i-1}}, \qquad i = L, \dots, 1 \,. \tag{B.3}$$

Here we use the notation $d_0 = D = |\mathbf{x}|$ and $d_L = d = |\mathbf{z}|$ and further assume $d_0 \geqslant d_1 \geqslant \dots \geqslant d_{L-1} \geqslant d_L$ which gives the bottleneck.

Assuming $\mathbf{x}$ is such that a corresponding latent variable $\mathbf{z}$ exists, i.e. that there exists $\mathbf{z}$ such that $\mathbf{x} = f(\mathbf{z})$, then we can write the likelihood of $\mathbf{x}$ through a stan-

Table B.2: Selection of most important hyperparameters and their setting. Convolutional kernels are square and latent dimensions are given without spatial dimensions which are given in the text. See appendix B.2 for more details.

Hyperparameter	Setting/Range
All	
Optimization	Adam [342]
Learning rate	$3e-4$
Batch size	128
Epochs	2000
Free bits	2 nats per $\mathbf{z}_i$ shared across latent dim.
Free bits constant	200 epochs
Free bits annealed	200 epochs
Activation	ReLU
Initialization	Data-dependent [574]
HVAE	
Latent dimensionality	128-64-32 (natural) / 8-16-8 (gray)
Convolution kernel	5-3-3
Stride	2-2-1
Warm up anneal period	200 epochs
BIVA	
Latent dimensionality	10-8-6 (spatial) 42-40-38-36-34-32-30 (dense)
Convolution kernel	5-3-3-3-3-3-3-3-3-3
Stride	2-1-1-2-1-2-1-1-1-1

dard change of variables (similar to flow-based models),

$$p(\mathbf{x}) = p(\mathbf{z}) \prod_{i=1}^{L} \left(\sqrt{\det \mathbf{J}_i^\mathsf{T} \mathbf{J}_i} \right)^{-1} , \tag{B.4}$$

where $\mathbf{J}_i$ is the Jacobian of f_i, i.e.

$$\mathbf{J}_i = \frac{\partial f_i}{\partial \mathbf{z}_i} \in \mathbb{R}^{d_i \times d_{i-1}} . \tag{B.5}$$

Here we use the notation that $\mathbf{z}_i$ is the representation at layer i. Note that $\mathbf{J}_i^\mathsf{T} \mathbf{J}_i$ is a $d_{i-1} \times d_{i-1}$ symmetric positive semidefinite matrix (determinant $\geqslant 0$).

The log-likelihood can be written as

$$\log p(\mathbf{x}) = \log p(\mathbf{z}) - \frac{1}{2} \sum_{i=1}^{L} \log \det \mathbf{J}_i^\mathsf{T} \mathbf{J}_i \ . \tag{B.6}$$

By construction of determinants, we can generally expect these determinants to grow with the dimensionality of the matrix. We should expect the determinant of a $d \times d$ matrix to be of the order $O(\lambda^d)$ for some number $\lambda > 0$. With that in mind, we should generally expect that

$$\det \mathbf{J}_{i+1}^\mathsf{T} \mathbf{J}_{i+1} < \det \mathbf{J}_i^\mathsf{T} \mathbf{J}_i \ , \tag{B.7}$$

due to the bottleneck assumption. If so, we see that the marginal likelihood $p(\mathbf{x})$ will be dominated by $\left(\sqrt{\det \mathbf{J}_1^\mathsf{T} \mathbf{J}_1} \right)^{-1}$, i.e. low-level features have a higher influence on the likelihood than more important semantic ones.

B.3.2   THE GAUSSIAN CASE

The previous remarks can be made more precise if we make distributional assumptions on the Jacobians. Here we will assume that the Jacobians of each layer follow a Gaussian distribution. Specifically, we will assume that each entry in $\mathbf{J}_i$ is distributed as $\mathcal{N}(0, \sigma^2)$. The analysis below extends to nonzero means and more general covariance structure, but this comes with a cost of less transparent notation. In this setting, $\mathbf{J}_i^\mathsf{T} \mathbf{J}_i$ follows a Wishart distribution (in the general setting it would follow a non-central Wishart distribution). Muirhead [468] tells us that the expected multiplicative contribution to the likelihood of each layer is

$$\mathbb{E}\left[\left(\sqrt{\det \mathbf{J}_i^\mathsf{T} \mathbf{J}_i} \right)^{-1} \right] = \sigma^{-d_{i-1}} 2^{-\frac{d_{i-1}}{2}} \frac{\Gamma_{d_{i-1}}\left(\frac{1}{2} d_i - \frac{1}{2} \right)}{\Gamma_{d_{i-1}}\left(\frac{1}{2} d_i \right)}$$

$$= \sigma^{-d_{i-1}} 2^{-\frac{d_{i-1}}{2}} \frac{\Gamma\left(\frac{1}{2}(d_i - d_{i-1}) \right)}{\Gamma\left(\frac{1}{2} d_i \right)} \tag{B.8}$$

where Γ_d is the multivariate Gamma function. Assuming that the increase in layer dimension $d_i - d_{i-1}$ is constant, then we see that (B.8) goes to zero as d_i goes to infinity as the Γ function grows super-exponentially to infinity. This super-exponential growth further implies that the first layers dominate the marginal likelihood $p(\mathbf{x})$. This is also visually evident in figure B.1.
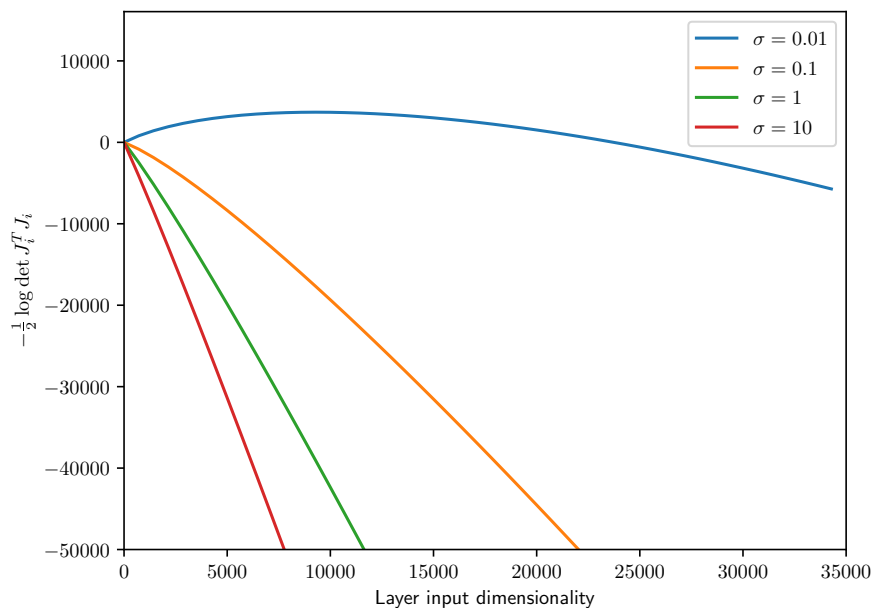
Figure B.1: The expected inverse volume change for Gaussian Jacobians (B.8) on a log-scale.

B.4   DERIVATION OF THE $\mathcal{L}^{>k}$ BOUND

In this section we present the derivation of $\mathcal{L}^{>k}$ and show that it is a lower bound on the marginal likelihood.

First, we consider a two-layered VAE with bottom-up inference. We proceed very similarly to the derivation of the regular ELBO and also use Jensen's inequal-

ity.

$$\log p(\mathbf{x}) = \log \int \int p(\mathbf{x}|\mathbf{z}_1)p(\mathbf{z}_1|\mathbf{z}_2)p(\mathbf{z}_2)d\mathbf{z}_1 d\mathbf{z}_2 \tag{B.9}$$

$$= \log \int \int \frac{q(\mathbf{z}_2|\mathbf{x})}{q(\mathbf{z}_2|\mathbf{x})} p(\mathbf{x}|\mathbf{z}_1)p(\mathbf{z}_1|\mathbf{z}_2)p(\mathbf{z}_2)d\mathbf{z}_1 d\mathbf{z}_2$$

$$= \log \int \int q(\mathbf{z}_2|\mathbf{x})p(\mathbf{z}_1|\mathbf{z}_2)\frac{p(\mathbf{x}|\mathbf{z}_1)p(\mathbf{z}_2)}{q(\mathbf{z}_2|\mathbf{x})}d\mathbf{z}_1 d\mathbf{z}_2$$

$$\geqslant \mathbb{E}_{p(\mathbf{z}_1|\mathbf{z}_2)q(\mathbf{z}_2|\mathbf{x})}\left[\log \frac{p(\mathbf{x}|\mathbf{z}_1)p(\mathbf{z}_2)}{q(\mathbf{z}_2|\mathbf{x})}\right] \equiv \mathcal{L}^{>1} \, .$$

Here, we have introduced the variational distribution $q(\mathbf{z}_2|\mathbf{x})$ which, naively, is different from any of the available variational distributions $q(\mathbf{z}_1|\mathbf{x})$ and $q(\mathbf{z}_2|\mathbf{z}_1)$. However, it's easy to see that we can simply define $q(\mathbf{z}_2|\mathbf{x}) = q(\mathbf{z}_2|d_1(\mathbf{x}))$ where $d_1(\mathbf{x}) = \mathbb{E}\left[q(\mathbf{z}_1|\mathbf{x})\right]$. I.e. we compute the distribution over $\mathbf{z}_2$ via the mode of $q(\mathbf{z}_1|\mathbf{x})$. This is possible since we exclusively manipulate the variational proposal distribution without altering the generative model $p(\mathbf{x}, \mathbf{z})$.

In general, the derivation of $\mathcal{L}^{>k}$ for an L-layered hierarchical VAE with $\mathbf{z} = \mathbf{z}_1, \ldots, \mathbf{z}_L$ is as follows:

$$\log p(\mathbf{x}) = \log \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \tag{B.10}$$

$$= \log \int \frac{q(\mathbf{z}_{>k}|\mathbf{x})}{q(\mathbf{z}_{>k}|\mathbf{x})} p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

$$= \log \int q(\mathbf{z}_{>k}|\mathbf{x})p(\mathbf{z})\frac{p(\mathbf{x}|\mathbf{z})}{q(\mathbf{z}_{>k}|\mathbf{x})}d\mathbf{z}$$

$$= \log \int q(\mathbf{z}_{>k}|\mathbf{x})p(\mathbf{z}_{\leqslant k}|\mathbf{z}_{>k})p(\mathbf{z}_{>k})\frac{p(\mathbf{x}|\mathbf{z})}{q(\mathbf{z}_{>k}|\mathbf{x})}d\mathbf{z}$$

$$= \log \int q(\mathbf{z}_{>k}|\mathbf{x})p(\mathbf{z}_{\leqslant k}|\mathbf{z}_{>k})\frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z}_{>k})}{q(\mathbf{z}_{>k}|\mathbf{x})}d\mathbf{z}$$

$$\geqslant \mathbb{E}_{p(\mathbf{z}_{\leqslant k}|\mathbf{z}_{>k})}\left[\log q(\mathbf{z}_{>k}|\mathbf{x})\frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z}_{>k})}{q(\mathbf{z}_{>k}|\mathbf{x})}\right]$$

$$\geqslant \mathbb{E}_{p(\mathbf{z}_{\leqslant k}|\mathbf{z}_{>k})q(\mathbf{z}_{>k}|\mathbf{x})}\left[\log \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z}_{>k})}{q(\mathbf{z}_{>k}|\mathbf{x})}\right] \equiv \mathcal{L}^{>k} \, .$$

Similar to the $L = 2$ case above, we have defined

$$q(\mathbf{z}_{>k}|\mathbf{x}) = q(\mathbf{z}_{>k}|d_k(\mathbf{x}))$$

with d_k defined recursively as

$$d_k(\mathbf{x}) = \mathbb{E}\left[q(\mathbf{z}_k|d_{k-1}(\mathbf{x}))\right], \qquad d_0(\mathbf{x}) = \mathbf{x} \, .$$

That is, we simply consider the inference network below z_{k+1} to be a deterministic encoder and forward pass the mode of each preceding variational distribution.

Additionally, we obtain $p(z_{\leqslant k}|z_{>k})p(z_{>k})$ by splitting

$$p(z) = p(z_L)p(z_{L-1}|z_L) \cdots p(z_1|z_2)$$

at index k. Importantly, we then evaluate

$$p(z_{>k}) = p(z_L)p(z_{L-1}|z_L) \cdots p(z_{k+1}|z_{k+2})$$

with samples from $q(z_{>k}|x)$ while

$$p(z_{\leqslant k}|z_{>k}) = p(z_k|z_{k+1})p(z_{k-1}|z_k) \cdots p(z_1|z_2)$$

is evaluated for z_k with $z_{k+1} \sim q(z_{>k}|x)$ and for $z_{<k}$ with $z_{>k}$ obtained conditionally from itself.

B.5   THE COMPLEMENTARY $\mathcal{L}^{<l}$ BOUND

We can generalize the $\mathcal{L}^{>k}$ bound by introducing the flipped version, $\mathcal{L}^{<l}$, which compared to $\mathcal{L}^{>k}$, instead samples the $L - l$ *highest* latent variables in the hierarchy from the prior $z_l, \dots, z_L \sim p_\theta(z_{\geqslant l}) = p_\theta(z_l|z_{l+1}) \cdots p_\theta(z_L)$ and the remaining lower latents from the approximate posterior $\hat{z}_1, \dots, \hat{z}_{l-1} \sim q_\phi(z_{<l}|x) = q_\phi(z_1|x)q_\phi(z_2|z_1) \cdots q_\phi(z_{l-1}|z_{l-2})$,

$$\mathcal{L}^{<l} = \mathbb{E}_{p_\theta(z_{\geqslant l})q_\phi(z_{<l}|x)} \left[\log \frac{p_\theta(x,z)p_\theta(z_{<l})}{q_\phi(z_{<l}|x)} \right]. \tag{B.11}$$

Similar to $\mathcal{L}^{>k}$, we recover the regular ELBO for $l = L$. Contrary to $\mathcal{L}^{>k}$, this bound puts as much emphasis on the lowest latent variables as the regular ELBO but keeps track of large deviation from the unconditional prior in the top $L-l$ KL-terms since it is not guided by the approximate posterior for $z_{>l}$. We hypothesize that this bound might be useful for OOD detection in cases where the discriminating factor is to be found in low-level statistics rather than high-level features.

Additionally, we can incorporate it in a generalized log likelihood-ratio between $\mathcal{L}^{<l}$ and $\mathcal{L}^{>k}$

$$LLR_{<l}^{>k} = \mathcal{L}^{<l} - \mathcal{L}^{>k}. \tag{B.12}$$

We hypothesize that this score, or the other possible permutations of it, might be useful for OOD detection but leave further examination to future work.

B.6   Note on the KL-term of hierarchical VAEs

In this research we choose model parameterizations relying on bottom-up inference [69],

$$q_\phi(\mathbf{z}|\mathbf{x}) = q_\phi(\mathbf{z}_1|\mathbf{x}) \prod_{i=2}^{L} q_\phi(\mathbf{z}_i|\mathbf{z}_{i-1}) \,. \tag{B.13}$$

We do this because bottom-up inference enables the model to learn covariance between the latent variables in the hierarchy. In the inference model, any latent variable is dependent on the latent variables below it in the hierarchy and, importantly, the top most latent variable is dependent on all other latent variables.

In contrast, a top-down inference model [614] has a topmost latent variable $\mathbf{z}_L$ that is independent of the other latent variables and is directly given by $\mathbf{x}$.

$$q_\phi(\mathbf{z}|\mathbf{x}) = q_\phi(\mathbf{z}_L|\mathbf{x}) \prod_{i=L-1}^{1} q_\phi(\mathbf{z}_i|\mathbf{z}_{i+1}) \,. \tag{B.14}$$

This, in essence, makes $\mathbf{z}_L$ a mean-field approximation without any covariance structure tying it to the other latent variables, $\mathrm{Cov}(z_{L,i}, z_{k,j}) = 0$ for $k < L$. Furthermore, since the approximate posterior (and the prior) typically have diagonal covariance, $\mathbf{z}_L$ is also mean-field within its own elements, $\mathrm{Cov}(z_{L,i}, z_{L,j}) = 0$ for $i \neq j$.

We hypothesize that the covariance of latent variables towards the top of the hierarchy with other latent variables is important for learning semantic representations. However, top-down inference models are easier to optimize as has recently been demonstrated [102, 614, 657].

In the following, we inspect the differences between the ELBO used for bottom-up inference and the ELBO used for top-down inference and show that it is not generally possible to decompose the total KL-divergence into separate KL-divergences per latent variable. Specifically, for top-down inference it is possible to obtain KL-divergence at the top-most latent variable and an expectation of a KL-divergence for the other latent variables. For bottom-up inference, the resulting terms are no longer KL-divergences except at the top-most latent variable.

We ask the question whether models relying on top-down inference are impeded in their use for semantic OOD detection, or whether they still learn to assign a more semantic representation in the top-most variables simply due to the flexibility of the deterministic neural network layers. This remains an open research question.

B.6.1 BOTTOM-UP INFERENCE

By splitting up the expectation, we can write the ELBO of a two-layer bottom-up hierarchical VAE as

$$
\log p(\mathbf{x}) \geqslant \mathbb{E}_{q(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})} \left[\log p(\mathbf{x}|\mathbf{z}_1) \right] \tag{B.15}
$$
$$
+ \mathbb{E}_{q(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})} \left[\log p(\mathbf{z}_1|\mathbf{z}_2) - \log q(\mathbf{z}_1|\mathbf{x}) \right]
$$
$$
+ \mathbb{E}_{q(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})} \left[\log p(\mathbf{z}_2) - \log q(\mathbf{z}_2|\mathbf{z}_1) \right] \; .
$$

We can write out the expectations in order to derive the KL-divergence terms of the bottom-up ELBO:

$$
\log p(\mathbf{x}) \geqslant \int \int \log p(\mathbf{x}|\mathbf{z}_1) d\mathbf{z}_2 \mathbf{z}_1 \tag{B.16}
$$
$$
+ \int q(\mathbf{z}_1|\mathbf{x}) \int q(\mathbf{z}_2|\mathbf{z}_1) \log \frac{p(\mathbf{z}_1|\mathbf{z}_2)}{q(\mathbf{z}_1|\mathbf{x})} d\mathbf{z}_2 \mathbf{z}_1
$$
$$
+ \int q(\mathbf{z}_1|\mathbf{x}) \int q(\mathbf{z}_2|\mathbf{z}_1) \log \frac{p(\mathbf{z}_2)}{q(\mathbf{z}_2|\mathbf{z}_1)} d\mathbf{z}_2 \mathbf{z}_1 \; .
$$

From the above, we can see that since the decomposition is in a reverse order, we cannot derive the KL-divergence for the second term. This will hold in general for L-layered models for any latent variables $\mathbf{z}_1, ..., \mathbf{z}_{L-1}$:

$$
\log p(\mathbf{x}) \geqslant \mathbb{E}_{q(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})} \left[\log p(\mathbf{x}|\mathbf{z}_1) \right] \tag{B.17}
$$
$$
+ \mathbb{E}_{q(\mathbf{z}_1|\mathbf{x})} \left[\mathbb{E}_{q(\mathbf{z}_2|\mathbf{z}_1)} \left[\log \frac{p(\mathbf{z}_1|\mathbf{z}_2)}{q(\mathbf{z}_1|\mathbf{x})} \right] \right]
$$
$$
+ \mathbb{E}_{q(\mathbf{z}_1|\mathbf{x})} \left[-D_{KL}[q(\mathbf{z}_2|\mathbf{z}_1) \parallel p(\mathbf{z}_2)] \right] \; .
$$

B.6.2 TOP-DOWN INFERENCE

By splitting up the expectation, we can write the ELBO of a two-layer top-down hierarchical VAE as

$$
\log p(\mathbf{x}) \geqslant \mathbb{E}_{q(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})} \left[\log p(\mathbf{x}|\mathbf{z}_1) \right] \tag{B.18}
$$
$$
+ \mathbb{E}_{q(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})} \left[\log p(\mathbf{z}_2|\mathbf{x}) - \log q(\mathbf{z}_2|\mathbf{x}) \right]
$$
$$
+ \mathbb{E}_{q(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})} \left[\log p(\mathbf{z}_1|\mathbf{z}_2) - \log q(\mathbf{z}_1|\mathbf{z}_2) \right] \; .
$$

We can write out the expectations in order to derive the KL-divergence terms:

$$\log p(\mathbf{x}) \geqslant \int \int \log p(\mathbf{x}|\mathbf{z}_1) d\mathbf{z}_1 \mathbf{z}_2 \tag{B.19}$$
$$+ \int q(\mathbf{z}_2|\mathbf{x}) \log \frac{p(\mathbf{z}_2|\mathbf{x})}{q(\mathbf{z}_2|\mathbf{x})} d\mathbf{z}_2$$
$$+ \int q(\mathbf{z}_2|\mathbf{x}) \int q(\mathbf{z}_1|\mathbf{z}_2) \log \frac{p(\mathbf{z}_1|\mathbf{z}_2)}{q(\mathbf{z}_1|\mathbf{z}_2)} d\mathbf{z}_1 \mathbf{z}_2 .$$

The KL-divergence terms can now easily be computed by:

$$\log p(\mathbf{x}) \geqslant \mathbb{E}_{q(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})} \left[\log p(\mathbf{x}|\mathbf{z}_1) \right] \tag{B.20}$$
$$- D_{KL}[q(\mathbf{z}_2|\mathbf{x}) \| p(\mathbf{z}_2)]$$
$$- \mathbb{E}_{q(\mathbf{z}_2|\mathbf{x})} [D_{KL}[q(\mathbf{z}_1|\mathbf{z}_2) \| p(\mathbf{z}_1|\mathbf{z}_2)]] .$$

Note that the KL-divergence in the second layer is not exact since it is dependent on the sample-noise from the layer below. An exact solution can only be derived if the latent variables $\mathbf{z}$ are all conditionally independent. However, this comes at the cost of not learning a covariance structure.

B.7 Additional results

We provide additional results for a model trained on FashionMNIST in table B.5, a model trained on MNIST in table B.6, a model trained on CIFAR10 in table B.4 and a model trained on SVHN in table B.3.

We note that while the likelihood is highly unreliable across the datasets, the proposed log likelihood-ratio score is consistent and always allows correct OOD detection with high AUROC↑.

Table B.3: Additional results for the HVAE model trained on SVHN. All results computed with 1000 importance samples.

OOD dataset	Metric	AUROC↑	AUPRC↑	FPR80↓
Trained on SVHN				
CIFAR10	$\mathcal{L}^{>0}$	0.992	0.993	0.004
CIFAR10	$\mathcal{L}^{>1}$	0.988	0.990	0.002
CIFAR10	$\mathcal{L}^{>2}$	0.746	0.756	0.468
CIFAR10	$LLR^{>1}$	0.939	0.950	0.052
SVHN	$\mathcal{L}^{>0}$	0.599	0.587	0.702
SVHN	$\mathcal{L}^{>1}$	0.555	0.543	0.755
SVHN	$\mathcal{L}^{>2}$	0.403	0.431	0.869
SVHN	$LLR^{>1}$	0.489	0.484	0.799

Table B.4: Additional results for the HVAE model trained on CIFAR10. All results computed with 1000 importance samples.

OOD dataset	Metric	AUROC↑	AUPRC↑	FPR80↓
Trained on CIFAR10				
SVHN	$\mathcal{L}^{>0}$	0.083	0.318	0.974
SVHN	$\mathcal{L}^{>1}$	0.097	0.320	0.972
SVHN	$\mathcal{L}^{>2}$	0.693	0.725	0.599
SVHN	$LLR^{>2}$	0.811	0.837	0.394
CIFAR10	$\mathcal{L}^{>0}$	0.485	0.488	0.817
CIFAR10	$\mathcal{L}^{>1}$	0.467	0.476	0.822
CIFAR10	$\mathcal{L}^{>2}$	0.411	0.433	0.869
CIFAR10	$LLR^{>1}$	0.469	0.479	0.835

Table B.5: Additional results for the HVAE model trained on FashionMNIST. All results computed with 1000 importance samples.

OOD dataset	Metric	AUROC↑	AUPRC↑	FPR80↓
Trained on FashionMNIST				
MNIST	$\mathcal{L}^{>0}$	0.268	0.363	0.882
MNIST	$\mathcal{L}^{>1}$	0.593	0.591	0.658
MNIST	$\mathcal{L}^{>2}$	0.712	0.750	0.548
MNIST	$LLR^{>1}$	0.986	0.987	0.011
notMNIST	$\mathcal{L}^{>0}$	0.916	0.932	0.116
notMNIST	$\mathcal{L}^{>1}$	0.983	0.986	0.000
notMNIST	$\mathcal{L}^{>2}$	0.997	0.997	0.000
notMNIST	$LLR^{>1}$	0.998	0.998	0.000
KMNIST	$\mathcal{L}^{>0}$	0.690	0.694	0.554
KMNIST	$\mathcal{L}^{>1}$	0.835	0.863	0.359
KMNIST	$\mathcal{L}^{>2}$	0.844	0.875	0.339
KMNIST	$LLR^{>1}$	0.974	0.977	0.017
Omniglot28x28	$\mathcal{L}^{>0}$	0.898	0.837	0.166
Omniglot28x28	$\mathcal{L}^{>1}$	0.991	0.989	0.011
Omniglot28x28	$\mathcal{L}^{>2}$	1.000	1.000	0.000
Omniglot28x28	$LLR^{>2}$	1.000	1.000	0.000
Omniglot28x28Inverted	$\mathcal{L}^{>0}$	0.261	0.361	0.879
Omniglot28x28Inverted	$\mathcal{L}^{>1}$	0.450	0.431	0.709
Omniglot28x28Inverted	$\mathcal{L}^{>2}$	0.557	0.574	0.678
Omniglot28x28Inverted	$LLR^{>1}$	0.954	0.954	0.050
SmallNORB28x28	$\mathcal{L}^{>0}$	0.982	0.984	0.000
SmallNORB28x28	$\mathcal{L}^{>1}$	0.998	0.998	0.000
SmallNORB28x28	$\mathcal{L}^{>2}$	1.000	1.000	0.000
SmallNORB28x28	$LLR^{>2}$	0.999	0.999	0.002
SmallNORB28x28Inverted	$\mathcal{L}^{>0}$	0.965	0.971	0.000
SmallNORB28x28Inverted	$\mathcal{L}^{>1}$	0.997	0.992	0.000
SmallNORB28x28Inverted	$\mathcal{L}^{>2}$	0.981	0.985	0.000
SmallNORB28x28Inverted	$LLR^{>2}$	0.941	0.946	0.069
FashionMNIST	$\mathcal{L}^{>0}$	0.476	0.484	0.816
FashionMNIST	$\mathcal{L}^{>1}$	0.475	0.482	0.817
FashionMNIST	$\mathcal{L}^{>2}$	0.475	0.484	0.823
FashionMNIST	$LLR^{>1}$	0.488	0.496	0.811

Table B.6: Additional results for the HVAE model trained on MNIST. All results computed with 1000 importance samples.

OOD dataset	Metric	AUROC↑	AUPRC↑	FPR80↓
Trained on MNIST				
FashionMNIST	$\mathcal{L}^{>0}$	1.000	1.000	0.000
FashionMNIST	$\mathcal{L}^{>1}$	1.000	1.000	0.000
FashionMNIST	$\mathcal{L}^{>2}$	0.981	0.983	0.003
FashionMNIST	$LLR^{>1}$	0.999	0.999	0.000
notMNIST	$\mathcal{L}^{>0}$	1.000	1.000	0.000
notMNIST	$\mathcal{L}^{>1}$	1.000	1.000	0.000
notMNIST	$\mathcal{L}^{>2}$	1.000	1.000	0.000
notMNIST	$LLR^{>1}$	1.000	0.999	0.000
KMNIST	$\mathcal{L}^{>0}$	1.000	1.000	0.000
KMNIST	$\mathcal{L}^{>1}$	1.000	1.000	0.000
KMNIST	$\mathcal{L}^{>2}$	0.987	0.987	0.011
KMNIST	$LLR^{>1}$	0.999	0.999	0.000
Omniglot28x28	$\mathcal{L}^{>0}$	1.000	1.000	0.000
Omniglot28x28	$\mathcal{L}^{>1}$	1.000	1.000	0.000
Omniglot28x28	$\mathcal{L}^{>2}$	1.000	1.000	0.000
Omniglot28x28	$LLR^{>1}$	1.000	1.000	0.000
Omniglot28x28Inverted	$\mathcal{L}^{>0}$	0.862	0.902	0.205
Omniglot28x28Inverted	$\mathcal{L}^{>1}$	0.923	0.943	0.056
Omniglot28x28Inverted	$\mathcal{L}^{>2}$	0.749	0.691	0.411
Omniglot28x28Inverted	$LLR^{>1}$	0.944	0.953	0.057
SmallNORB28x28	$\mathcal{L}^{>0}$	1.000	1.000	0.000
SmallNORB28x28	$\mathcal{L}^{>1}$	1.000	1.000	0.000
SmallNORB28x28	$\mathcal{L}^{>2}$	1.000	1.000	0.000
SmallNORB28x28	$LLR^{>1}$	1.000	1.000	0.000
SmallNORB28x28Inverted	$\mathcal{L}^{>0}$	1.000	1.000	0.000
SmallNORB28x28Inverted	$\mathcal{L}^{>1}$	1.000	1.000	0.000
SmallNORB28x28Inverted	$\mathcal{L}^{>2}$	0.977	0.980	0.001
SmallNORB28x28Inverted	$LLR^{>1}$	0.985	0.987	0.000
MNIST	$\mathcal{L}^{>0}$	0.488	0.486	0.807
MNIST	$\mathcal{L}^{>1}$	0.469	0.469	0.816
MNIST	$\mathcal{L}^{>2}$	0.514	0.505	0.791
MNIST	$LLR^{>2}$	0.515	0.507	0.792

Supplementary material: Benchmarking generative latent variable models for speech

C.1 Reproducibility statement

The source code used for the work presented in this paper will be made available before the conference. This code provides all details, practical and otherwise, needed to reproduce the results in this paper including data preprocessing, model training, model likelihood and latent space evaluation. The source code also includes scripts for downloading and preparing the LibriSpeech, LibriLight and TIMIT datasets. The LibriSpeech and LibriLight datasets are open source and can be downloaded with the preparation scripts. They are also available at `https://www.openslr.org/12` and `https://github.com/facebookresearch/libri-light`, respectively. The TIMIT dataset is commercial and must be purchased and downloaded from `https://catalog.ldc.upenn.edu/LDC93S1` before running the preparation script.

The stochastic latent variable models considered in this work do not provide an exact likelihood estimate nor an exact latent space representation. For the likelihood, they provide a stochastic lower bound and some variation in the reproduced likelihoods as well as latent representations must be expected between otherwise completely identical forward passes. This variance is fairly small in practice when averaging over large datasets such as those considered in this work. We seed our experiments to reduce the randomness to a minimum, but parts of the algorithms underlying the CUDA framework are stochastic for efficiency. To retain computational feasibility, we do not run experiments with a deterministic CUDA backend.

C.2 Ethics statement

The work presented here fundamentally deals with automated perception of speech and generation of speech. These applications of machine learning potentially raise a number of ethical concerns. For instance, the these models might see possibly adverse use in automated surveillance and generation of deep fakes. To counter some of these effects, this work has focused on openness by using publicly available datasets for model development and benchmarking. Additionally, the work will open source the source code used to create these results. Ensuring the net positive effect of the development of these technologies is and must continue to be an ongoing effort.

We do not associate any significant ethical concerns with the datasets used in this work. However, one might note that the TIMIT dataset has somewhat skewed distributions in terms of gender and race diversity. Specifically, the male to female ratio is about two to one while the vast majority of speakers are Caucasian. Such statistics might have an effect of some ethical concern on downstream applications derived from such a dataset as also highlighted in recent research [351]. In LibriSpeech, there is an approximately equal number of female and male speakers while the diversity in race is unknown to the authors.

C.3   DATASETS

TIMIT   TIMIT [196] is a speech dataset which contains 16 kHz recordings of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences. It amounts to 6300 total recordings splits approximately in 3.94 hours of audio for training and 1.43 hours of audio for testing. No speakers or sentences in the test set are in the training set. The full train and test subsets of TIMIT are as in previous work [3, 126, 188]. We randomly sample 5% of the training set to use as a validation set. TIMIT includes temporally aligned annotations of phonemes and words as well as speaker metadata such as gender, height, age, race, education level and dialect region [196].

LibriSpeech and LibriLight   The LibriSpeech dataset [506] consists of readings of public domain audiobooks amounting to approximately 1000 h of audio. The data is derived from the LibriVox project. LibriLight [320] is a subset of LibriSpeech created as an automatic speech transcription (ASR) benchmark with limited or no supervision. We specifically train on the 100 h train-clean-100 subset of LibriSpeech and the 10 h subset of LibriLight. In all cases we evaluate on all the test splits dev-clean, dev-other, test-clean, test-other.

Both datasets represent the audio as 16 bit pulse code modulation (PCM) sampled at 16 000 Hz.

C.4   MODEL ARCHITECTURES

This section details model architectures. See appendix appendix C.10 for graphical models and appendix appendix C.5 for training details.

WaveNet   We implement WaveNet as described in the original work [496] but use a discretized mixture of logistics as the output distribution as also done in other work [501]. Our WaveNet is not conditioned on any signal other than the raw waveform. The model applies the causal convolution directly to the raw

waveform frames (i.e. one input channel). An alternative option that we did not examine is to replace the initial convolution with an embedding lookup with a learnable vector for each waveform frame value.

LSTM   The LSTM baseline uses an MLP encoder to embed the waveform sub-segment $x_{t:t+s-1}$ to a feature vector before feeding it to the LSTM cell. The encoder is similar to the parameterization of $\phi_{\text{vrnn}}^{\text{enc}}$ for the VRNN described above. The LSTM cell produces the hidden state d_t from $x_{t:t+s-1}$ and passes it to a decoder. Like the encoder, the decoder is parameterized like $\phi_{\text{vrnn}}^{\text{dec}}$ of the VRNN. It outputs the waveform predictions $x_{t+s:t+2s-1}$ from the hidden state d_t. The LSTM model uses a single vanilla unidirectional LSTM cell.

VRNN   We implement the VRNN as described in the original work [126] and verify that we can reproduce the original Gaussian likelihood TIMIT results. We replace the Gaussian output distribution with the DMoL.

SRNN   We implement the VRNN as described in the original work [188] and verify that we can reproduce the original Gaussian likelihood TIMIT results. We replace the Gaussian output distribution with the DMoL.

CW-VAE   We implement the CW-VAE based on the original work [577] but with some modifications also briefly described in section section 7.3.6. We replace the encoder/decoder model architectures of the original work with architectures designed for waveform modeling. Specifically, the encoder and decoder are based on the Conv-TasNet [430] and uses similar residual block structure. However, contrary to the Conv-TasNet, we require downsampling factors larger than two. In order to achieve this we use strides of two in the separable convolution of each block. With e.g. six blocks we hence get an overall stride of $2^6 = 64$. We can then add additional blocks with unit stride. We also need to modify the residual connections that skip strided convolutions. Specifically, we replace the residual with a single convolution with stride equal to the stride used in the separable convolution. This convolution uses no nonlinearity and hence simply learns a local linear downsampling.

STCN   We implement the STCN as described in the original work [3] and verify that we can reproduce the original Gaussian likelihood TIMIT results. We replace the Gaussian output distribution with the DMoL. We use the best-performing version of the STCN reported in the original paper, namely the "STCN-dense" variant which conditions the observed variable on all five latent variables in the hierarchy. For the ablation experiment, we remove the bottom four latent variables.

That is, we completely remove the corresponding four small densely connected networks that parameterize the prior and posterior distributions based on deterministic representations of the WaveNet encoder. We keep the top most prior and posterior networks and use them to parameterize a latent variable of 256. This maintains the widest bottleneck of the model as well as almost all of the model's capacity.

ASR model   The ASR model used for the phoneme recognition experiments is a three-layered bidirectional LSTM. We apply temporal dropout between the LSTM layers and also after the final layer. Temporal dropout works similar to regular dropout but samples the entries of the hidden state to mask only once and apply it to all timesteps, i.e. masking $\mathbf{h}_t$ at vector index i for all t (and i). We mask by zeroing vector elements. We never mask the first timestep. We apply temporal dropout with masking probability of 0.3 for the 3.7h subset, 0.35 for the 1h subset and 0.4 for the 10m subset. The only difference in model architecture between the evaluation of different representations is the first affine transformation; from the dimensionality of the representation to the hidden state size of the LSTM. This gives rise to a very small difference in model capacity and parameter count which we find is negligible. We set the hidden unit size to 256.

C.5   TRAINING DETAILS

Likelihood benchmark   We implement all models and training scripts in Py-Torch 1.9 [517]. For both datasets we use the Adam optimizer [342] with default parameters as given in PyTorch. We use learning rate $3e-4$ and no learning rate schedule. We use PyTorch automatic mixed precision (AMP) to significantly reduce memory consumption. We did not observe any significant difference in final model performance compared to full (32 bit) precision.

For us train stateful models (LSTM, VRNN, SRNN and CW-VAE) on the full sequence lengths padding batches with zeros when examples are not of equal length. We sample batches such that they consist of examples that are approximately the same length to minimize the amount of computation wasted on padding.

For $s = 1$, we train stateless models (WaveNet, STCN) on random subsegments of the training examples and resample every epoch. This reduces memory requirements but does not bias the gradient. The subsequences are chosen to be of length 16000 which is larger than the receptive fields of the models and corresponds to one second of audio in TIMIT and LibriSpeech. For $s = 64$ and $s = 256$ we train the stateless models on the full example lengths similar to the stateful models since the receptive field is effectively s times larger and the shorter sequence length reduces memory requirements.

In testing, we evaluate on the full sequences. Due to memory constrains, for LibriSpeech, we need to split the test examples into subsegments since the average sequence length in Librispeech is about 4 times longer than that of TIMIT. Hence, we do multiple forward passes per test example, one for each of several subsegments. We carry along the internal state for models that are autoregressive in training (LSTM, VRNN, SRNN, CW-VAE) and define segments to overlap according to model architecture.

Phoneme recognition   The ASR experiment consists of two stages: 1) pre-training of the unsupervised model and 2) training of the ASR model. The pre-training is done as for the likelihood benchmark above. The ASR model is trained using the Adam optimizer [342] with default parameters as given in PyTorch. We use learning rate $3e - 4$ and no learning rate schedule.

For the spectrogram, WaveNet and the LSTM, we extract the representation only once and train the ASR model on these. Since the models are deterministic and do not parameterize distributions, this is the only option. For the LVMs, we resample the latent representation of a training example at every epoch. This is the most principled approach as these models parameterize probability distributions. Furthermore, using a single sample would be subject to artificially high variance in the representations while it is not straightforward to establish a sound mean representation for sequential models.

C.6   Converting the likelihood to units of bits per frame

Here we briefly describe how to compute a likelihood in units of bits per frame (bpf). In the main text, we use log to mean $\log_e$, but here we will be explicit. In general, conversion from nats to bits (i.e., from $\log_e$ to $\log_2$) is achieved by $\log_2(x) = \log_e(x)/\log_2(e)$. Remember that $\log_2 p(\mathbf{x}_{1:T})$ generally factorizes as $\sum_t \log_2 p(x_t|\cdot)$. In sequence modeling, it is important to remember that each example $\mathbf{x}^i$ must be weighted differently according the sequence length of that specific example. This is in contrast to computing bits per dimension in the image domain where images in a dataset are usually of the same dimensions. Thus, we compute the log-likelihood in bits per frame over the entire dataset as

$$\mathcal{L}(\mathbf{x}^i) = \frac{1}{\sum_i T_i} \sum_i \sum_t \log_2 p(\mathbf{x}_t^i) \ , \tag{C.1}$$

where i denotes the example index, T_i is the length of example $\mathbf{x}^i$ in waveform frames and t is the time index. If a single timestep $\mathbf{x}_t^i$ represents multiple waveform frames stacked with some stack size s, it is important to note that the sum over t only has T_i/s elements. For the LVMs, the term $\log_2 p(\mathbf{x}_t^i)$ is lower bounded by the ELBO in (7.1).

C.7   ADDITIONAL LIKELIHOOD RESULTS

TIMIT, μ-law, DMoL   We provide additional results on TIMIT with audio represented as μ-law encoded PCM in table C.2. Details are as presented in the main paper.

TIMIT, linear, DMoL   : We provide results on TIMIT with audio represented as linear PCM (raw PCM) in table C.1. Except for the encoding, details are as for μ-law encoded TIMIT

TIMIT, linear, Gaussian   We also provide some results on TIMIT with the audio instead represented as linear PCM (linearly encoded) and using Gaussian output distributions as has been done previously in the literature [3, 126, 188, 364]. We use $s = 200$ for comparability to the previous work. We provide the results in table C.3 and include likelihoods reported in the literature for reference. For our models, we use the same architectures as before but replace the discretized mixture of logistics with either a Gaussian distribution or a mixture of Gaussian distributions.

We constrain the variance of the Gaussians used with our models to be at least $\sigma_{\min}^2 = 0.01^2$ in order to avoid the variance going to zero, the likelihood going to infinity and optimization becoming unstable. The Gaussian standard deviation is clamped at minimum 0.001 by [3].

From table C.3 we note that the performance of the CW-VAE with Gaussian output distribution when modeling linear PCM (i.e. not μ-law encoded) does not compare as favorably to the other baselines as it did with the discretized mixture of logistics distribution. We hypothesize that this has to do with using a Gaussian output distribution in latent variable models which, as has been reported elsewhere [446], leads to a likelihood function that is unbounded above and can grow arbitrarily high. We discuss this phenomenon in further detail in section appendix C.8.

We specifically hypothesize that models that are autoregressive in the observed variable (VRNN, SRNN, Stochastic WaveNet, STCN) are well-equipped to utilize local smoothness to put very high density on the correct next value and that this in turn leads to a high degree of exploitation of the unboundedness of the likelihood. Not being autoregressive in the observed variable, the CW-VAE cannot exploit this local smoothness in the same way. Instead, the reconstruction is conditioned on a stochastic latent variable, $p(\mathbf{x}_t | \mathbf{z}_t^1)$, which introduces uncertainty and likely larger reconstruction variances.

Table C.1: Model likelihoods on TIMIT represented as a 16 bit linear PCM. The STCN converges to a poor local minimum and sometimes diverges when modeling linear PCM with $s = 1$.

s	**Model**	**Configuration**	$\mathcal{L}$ **[bpf]**
1	Uniform	Uninformed	16.00
1	DMoL	Optimal	10.70
-	FLAC	Linear PCM	8.582
1	WaveNet	$D_C = 96$	**7.246**
1	LSTM	$D_d = 256, L = 1$	7.295
1	VRNN	$D_z = 256$	$\leqslant 7.316$
1	SRNN	$D_z = 256$	$\leqslant 7.501$
1	STCN	$D_z = 256, L = 5$	$\leqslant 9.970$
64	WaveNet	$D_c = 96$	8.402
64	LSTM	$D_d = 256, L = 1$	8.357
64	VRNN	$D_z = 256$	$\leqslant 8.103$
64	SRNN	$D_z = 256$	$\leqslant 8.036$
64	CW-VAE	$D_z = 96, L = 1$	$\leqslant 7.989$
64	STCN	$D_z = 256, L = 5$	$\leqslant$ **7.768**
256	WaveNet	$D_c = 96$	9.018
256	LSTM	$D_d = 256, L = 1$	8.959
256	VRNN	$D_z = 256$	$\leqslant 8.739$
256	SRNN	$D_z = 256$	$\leqslant 8.674$
256	CW-VAE	$D_z = 96, L = 1$	$\leqslant 8.406$
256	STCN	$D_z = 256, L = 5$	$\leqslant$ **8.196**

C.8 ADDITIONAL DISCUSSION ON GAUSSIAN LIKELIHOODS IN LVMs

As noted in section appendix C.7, we constrain the variance of the output distribution of our models to be $\sigma^2_{min} = 0.01^2$ for the additional results on TIMIT with Gaussian outputs. This limits the maximum value attainable by the prediction/reconstruction density of a single waveform frame x_t.

Specifically, we can see that since

$$\log p(x_t|\cdot) = \log \mathcal{N}\left(x_t; \mu_t, \max\left\{\sigma^2_{min}, \sigma^2_t\right\}\right) , \tag{C.2}$$

the best prediction/reconstruction density is achieved when $\sigma^2 \leqslant \sigma^2_{min}$ and $\mu = x_t$. Here $\cdot$ indicates any variables we might condition on such as the previous input frame x_{t-1} or some latent variables. We can evaluate this best case scenario

Table C.2: Model likelihoods on TIMIT represented as a 16 bit μ-law encoded PCM.

s	Model	Configuration	$\mathcal{L}$ [bpf]
1	WaveNet	$D_C = 16$	11.27
1	WaveNet	$D_C = 24$	11.14
1	WaveNet	$D_C = 32$	11.03
1	WaveNet	$D_C = 96$	10.88
1	WaveNet	$D_C = 128$	10.98
1	WaveNet	$D_C = 160$	10.91
1	LSTM	$D_d = 128, L = 1$	11.40
1	LSTM	$D_d = 256, L = 1$	11.11
1	VRNN	$D_z = 256$	$\leqslant 11.09$
1	SRNN	$D_z = 256$	$\leqslant 11.19$
1	STCN	$D_z = 256, L = 5$	$\leqslant 11.77$
4	LSTM	$D_d = 256, L = 1$	11.65
16	LSTM	$D_d = 256, L = 1$	12.54
16	LSTM	$D_d = 256, L = 2$	12.54
16	LSTM	$D_d = 256, L = 3$	12.44
64	WaveNet	$D_c = 96$	13.30
64	LSTM	$D_d = 96, L = 1$	13.49
64	LSTM	$D_d = 96, L = 2$	13.46
64	LSTM	$D_d = 96, L = 3$	13.40
64	LSTM	$D_d = 256, L = 1$	13.27
64	LSTM	$D_d = 256, L = 2$	13.29
64	LSTM	$D_d = 256, L = 3$	13.31
64	LSTM	$D_d = 512, L = 1$	13.37
64	LSTM	$D_d = 512, L = 2$	13.37
64	LSTM	$D_d = 512, L = 3$	13.41
64	VRNN	$D_z = 96$	$\leqslant 12.93$
64	VRNN	$D_z = 256$	$\leqslant 12.54$
64	SRNN	$D_z = 96$	$\leqslant 12.87$
64	SRNN	$D_z = 256$	$\leqslant 12.42$
64	CW-VAE	$D_z = 96, L = 1$	$\leqslant 12.44$
64	CW-VAE	$D_z = 96, L = 2$	$\leqslant 12.17$
64	CW-VAE	$D_z = 96, L = 3$	$\leqslant 12.15$
64	CW-VAE	$D_z = 256, L = 2$	$\leqslant 12.10$
64	STCN	$D_z = 256, L = 1$	$\leqslant 12.32$
64	STCN	$D_z = 256, L = 5$	$\leqslant 11.78$
256	WaveNet	$D_c = 96$	14.11
256	LSTM	$D_d = 256, L = 1$	14.20
256	LSTM	$D_d = 256, L = 2$	14.17
256	LSTM	$D_d = 256, L = 3$	14.26
256	VRNN	$D_z = 96$	$\leqslant 13.51$
256	VRNN	$D_z = 256$	$\leqslant 13.27$
256	SRNN	$D_z = 96$	$\leqslant 13.28$
256	SRNN	$D_z = 256$	$\leqslant 13.14$
256	CW-VAE	$D_z = 96, L = 1$	$\leqslant 13.11$
256	CW-VAE	$D_z = 96, L = 2$	$\leqslant 12.97$
256	CW-VAE	$D_z = 96, L = 3$	$\leqslant 12.87$
256	STCN	$D_z = 256, L = 1$	$\leqslant 13.07$
256	STCN	$D_z = 256, L = 5$	$\leqslant 12.52$

Table C.3: Model likelihoods on TIMIT represented as globally normalized 16 bit linear PCM. Contrary to the other likelihoods reported in this paper, here they are given in units of nats and obtained by summing the likelihood over time and over all examples in the dataset and dividing by the total number of examples. In the table, Normal refers to using a Gaussian likelihood and GMM refers to using a Gaussian Mixture Model likelihood with 20 components. Models with asterisks * are our implementations while remaining results are as reported in the referenced work.

s	Model	Configuration	$\mathcal{L}$ [nats]
1	WaveNet	Normal	119656
1	WaveNet	GMM-2	120699
1	WaveNet	GMM-20	121681
200	WaveNet [3]	GMM-20	30188
200	WaveNet [3]	Normal	-7443
200	Stochastic WaveNet* [364]	Normal	$\geqslant$72463
200	VRNN [126]	Normal	$\approx$28982
200	SRNN [188]	Normal	$\geqslant$60550
200	STCN [3]	GMM-20	$\geqslant$69195
200	STCN [3]	Normal	$\geqslant$64913
200	STCN-dense [3]	GMM-20	$\geqslant$71386
200	STCN-dense [3]	Normal	$\geqslant$70294
200	STCN-dense-large [3]	GMM-20	$\geqslant$77438
200	CW-VAE*	$L = 1, D_z = 96$, Normal	$\geqslant$41629

for $\sigma^2_{min} = 0.01^2$,

$$\log \mathcal{N}\left(x_t; x_t, \sigma^2_{min}\right) = -\frac{1}{2} \log 2\pi - \frac{1}{2} \log \sigma^2_{min} - \frac{1}{2\sigma^2_{min}}(x_t - x_t)$$

$$= -\frac{1}{2} \log 2\pi - \frac{1}{2} \log 0.01^2$$

$$= 3.686 \ . \tag{C.3}$$

Hence, with perfect prediction/reconstruction and the minimal variance (0.01^2), a waveform frame contributes to the likelihood with 3.686 nats. With an average test set example length of 49 367.3 frames frames this leads to a best-case likelihood of 181967. We provide a list of maximally attainable Gaussian likelihoods on TIMIT for different minimal variances in table C.4. One can note that the maximal likelihood at $\sigma^2_{min} = 0.1^2$ is lower than the likelihoods achieved by

some models in table C.3. This indicates that the models learn to use very small variances in order to increase the likelihood. Empirically, standard deviations smaller than approximately 0.001 can result in numerical instability.

Table C.4: The highest possible Gaussian log-likelihoods (max $\mathcal{L}$) attainable on the TIMIT test set as computed by (C.2) with different values of the minimum variance $\sigma_{\min}^2$.

$\sigma_{\min}$	$\sigma_{\min}^2$	max $\mathcal{L}$
1	1	−45367
0.5	0.25	−11146
0.1	0.01	68307
0.05	0.0025	102525
0.01	0.0001	181979
0.005	0.000025	216198
0.001	0.000001	295651

C.9   ADDITIONAL DISCUSSION ON THE CHOICE OF OUTPUT DISTRIBUTION

The DMoL uses a discretization of the continuous logistic distribution to define a mixture model over a discrete random variable. This allows it to parameterize multimodal distributions which can express ambiguity about the value of $\mathbf{x}_t$. The model can learn to maximize likelihood by assigning a bit of probability mass to multiple potential values of $\mathbf{x}_t$.

While this is well-suited for autoregressive modeling, for which the distribution was developed, the potential multimodality poses a challenge for non-autoregressive latent variable models which independently sample multiple neighboring observations at the output. In fact, if multiple neighboring outputs defined by the subsequence $\mathbf{x}_{t_1:t_2}$ have multimodal $p(\mathbf{x}_t|\cdot)$, we risk sampling a subsequence where each neighboring value expresses different potential realities, independently.

Interestingly, most work on latent variable models with non-autoregressive output distributions seem to ignore this fact and simply employ the mixture distribution with 10 mixture components [102, 434, 657]. However, given the empirically good results of latent variable models for image generation, this seems to have posed only a minor problem in practice. We speculate that this is due to the high degree of similarity between neighbouring pixels in images. I.e. if

the neighboring pixels are nuances of red, then, in all likelihood, so is the central pixel.

In the audio domain, however, neighbouring waveform frames can take wildly different values, especially at low sample rates. Furthermore, waveforms exhibit a natural symmetry between positive and negative amplitudes. Hence, it seems plausible that multimodality may pose a larger problem in non-autoregressive speech generation by causing locally incoherent samples than it seems to do in image modelling.

C.10 ADDITIONAL GRAPHICAL MODELS

In figure C.1 we show the graphical model of the recurrent cell of the CW-VAE for a single timestep. As noted in [577], this cell is very similar to the one of the Recurrent State Space Model (RSSM) [230]. In figure C.2 we show the unrolled graphical models of a three-layered CW-VAE with $k_1 = 1$ and $c = 2$ yielding $k_2 = 2$ and $k_3 = 4$. We show both the generative and inference models and highlight in blue the parameter sharing between the two models due to top-down inference. In figure C.3 we show the graphical models of the STCN [3] at a single timestep. The model has three layers and shares the parameters of the WaveNet encoder between the inference and generative models. In figure C.4 we illustrate the unrolled graphical models of the inference and generative models of the VRNN [126]. We include the deterministic variable $\mathbf{d}_t$ in order to illustrate the difference to other latent variable models. Likewise, in figure C.5 we illustrate the unrolled graphical models the SRNN [188].

C.11 ADDITIONAL LATENT EVALUATION

We visualize the performance of a k-nearest-neighbour classifier for classification of speaker gender and height in figure C.6. The classifier is fitted to time-averaged latent representations and Mel-features. We divide the height into three classes: below 175 cm, above 185 cm and in-between. Compared to phonemes, the gender and height of a speaker are global attributes that affect the entire signal. In both cases, we see improved performance from using the learned latent space over Mel-features. Notably, $\mathbf{z}^2$ is outperformed by the Mel-features for gender identification which may indicate that $\mathbf{z}^2$ learns to ignore this attribute compared to $\mathbf{z}^1$.

We provide some additional latent space clustering of speaker gender in figure C.7 and of speaker height in figure C.8.

All results presented here are obtained with a 2-layered CW-VAE trained on μ-law encoded PCM similar to the one in table 7.1.
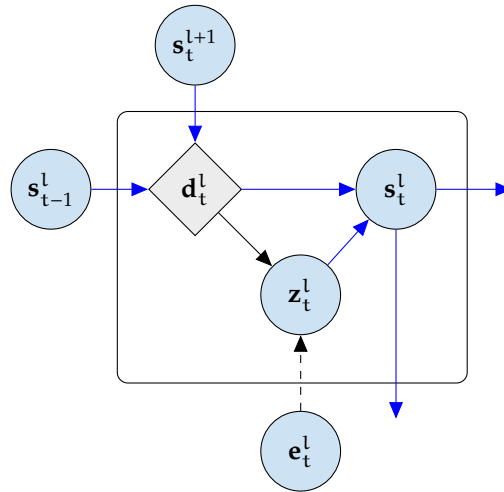
Figure C.1: CW-VAE cell state $\mathbf{s}_t^l$ update. The cell state is given as $\mathbf{s}_t^l = (\mathbf{z}_t^l, \mathbf{d}_t^l)$ where $\mathbf{d}_t^l$ is the deterministic hidden state of a Gated Recurrent Unit [105]. The vector $\mathbf{e}_t^l$ is computed from $\mathbf{x}_t$ by the encoder network which outputs L encodings, one for each latent variable, similar to that of a Ladder VAE [614]. All blue arrows are shared between generation and inference. The dashed arrow is used only during inference. The solid arrow has unique transformations during inference and generation.

C.12   DISTRIBUTION OF PHONEME DURATION IN TIMIT

In figure C.9 we plot a box plots of the duration of each phoneme in the TIMIT dataset. We do this globally as well as for a single speaker to show that phoneme duration can vary between individual speakers.

A description of the phonemes used for the TIMIT dataset can be found at `https://catalog.ldc.upenn.edu/docs/LDC93S1/PHONCODE.TXT`.

C.13   MODEL SAMPLES AND RECONSTRUCTIONS

We provide samples and reconstructions for some of the models considered here at the following URL: `https://doi.org/10.5281/zenodo.5911899`. The samples are generated from the prior of Clockwork VAE, SRNN and VRNN and from a WaveNet by conditioning on pure zeros. All models are configured as those reported in table 7.1. Importantly, the samples are unconditional. Hence, they are *not* reconstructions inferred from a given input nor are they conditioned
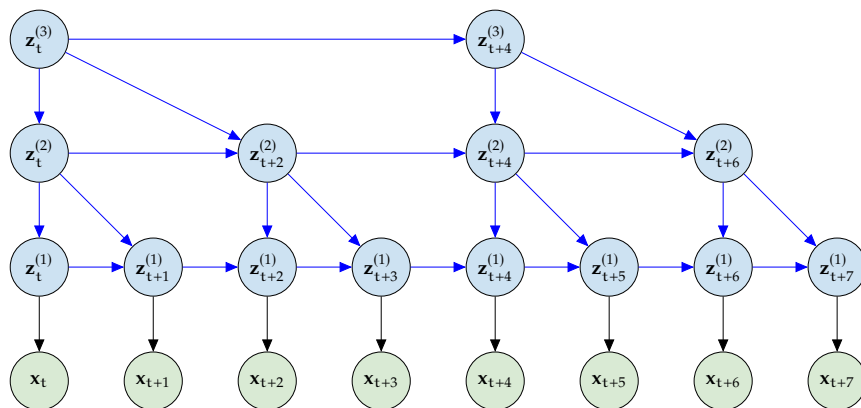
(a)
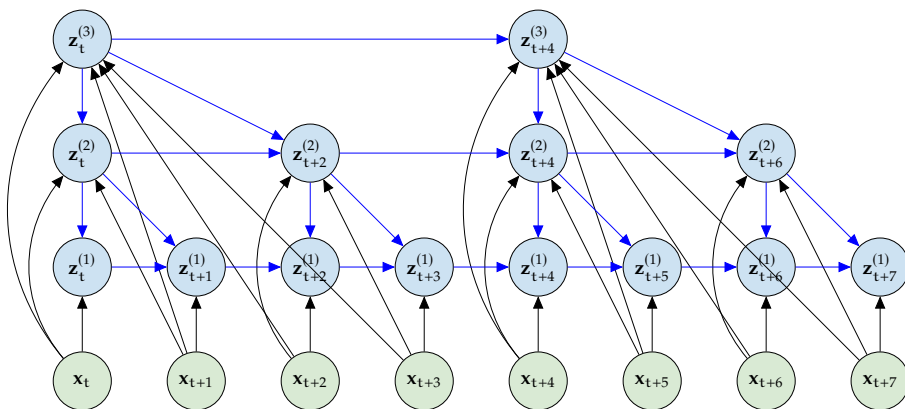
(b)

Figure C.2: CW-VAE [577] generative model $p(\mathbf{x}, \mathbf{z})$ in ((a)) and inference model $q(\mathbf{z}|\mathbf{x})$ in ((b)) for a three-layered model with $k_1 = 1$ and $c = 2$ giving $k_2 = 2$ and $k_3 = 4$ unrolled over eight steps in the observed variable. Blue arrows are (mostly) shared between the inference and generative models. See figure C.1 for a detailed graphical model expanding on the latent nodes $\mathbf{z}_t^l$ and parameter sharing.

(a)

(b)
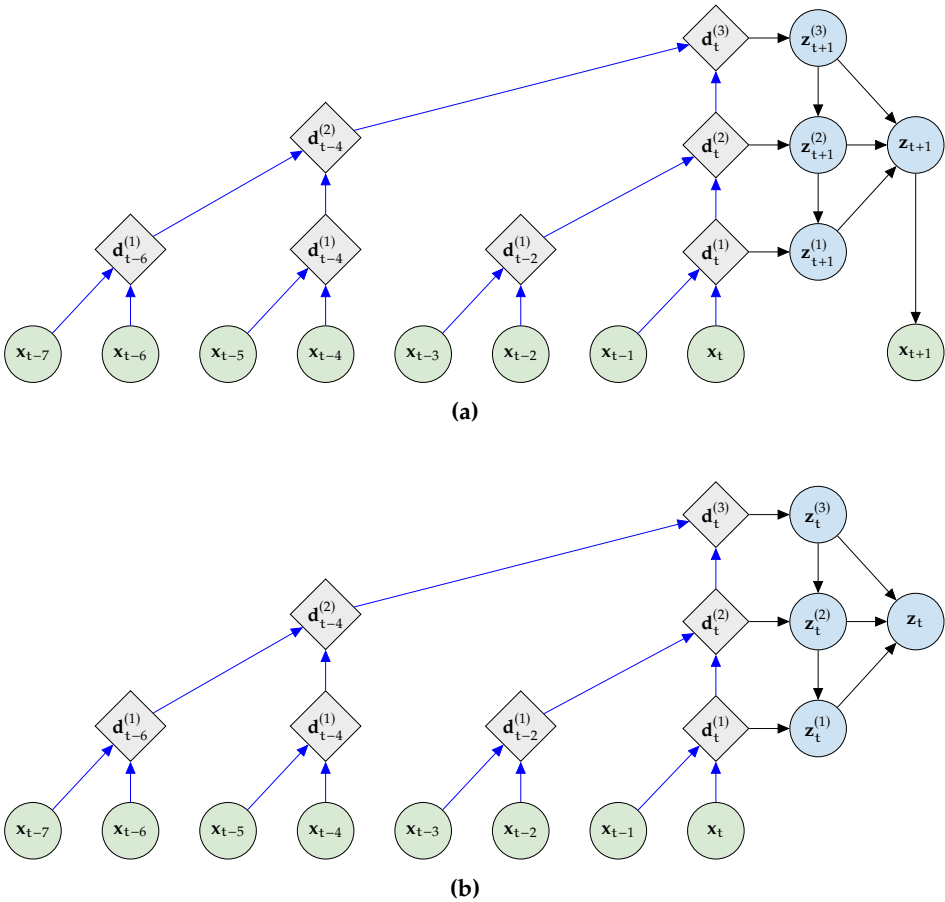
Figure C.3: STCN [3] generative model $p(\mathbf{x}, \mathbf{z})$ in ((a)) and inference model $q(\mathbf{z}|\mathbf{x})$ in ((b)) for a single time-step. The WaveNet autoregressive encoder is shared between generative and inference models. It is depicted here with only one stack of three layers in order to illustrate the dilated convolution with limited space. In practice, the model uses ten layers in each of five stacks/cycles resulting in a much larger receptive field. Importantly, the model parameterizes the five latent variables using the last deterministic representation $\mathbf{d}^{(1)}$ from each stack, i.e. only every fifth l starting from $l = 5$ and ending at $l = 25$. Note that the generative model uses the prior to transform the WaveNet hidden states $\mathbf{d}_t^{(1)}$ into the latent variable $\mathbf{z}_{t+1}^{(1)}$ one step ahead in time compared to the approximate posterior which infers $\mathbf{z}_t^{(1)}$. Also note that $\mathbf{z}_t$ is constructed by concatenating all $\mathbf{z}_t^{(1)}$. The original paper explores setting $\mathbf{z}_t$ equal to $\mathbf{z}_t^{(1)}$. The best-performing STCN for speech, which also the one we implement, uses a WaveNet decoder to predict $\mathbf{x}_{t+1}$ from a sequence of $\mathbf{z}_t$ rather than a per-timestep transform. Blue arrows are shared between the inference and generative models.
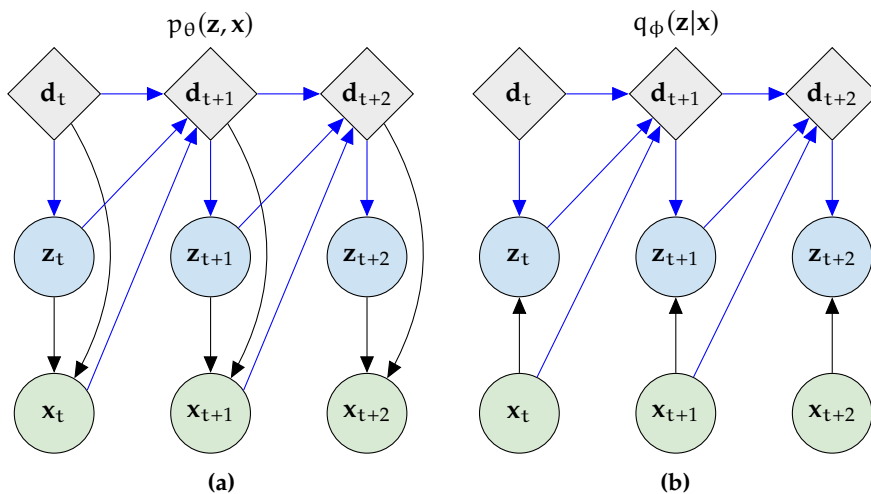
Figure C.4: VRNN [126] generative model $p(\mathbf{x}, \mathbf{z})$ in ((a)) and inference model $q(\mathbf{z}|\mathbf{x})$ in ((b)) unrolled over three steps in the observed variable. Blue arrows are shared between the inference and generative models.

on any auxiliary data like text.

Although sample quality is a somewhat subjective matter, we find the quality of the unconditional Clockwork VAE to be better than those of our VRNN and SRNN. WaveNet is known to produce samples with intelligible speech when conditioned on e.g. text, but unconditional samples from WaveNet lack semantic content such as words as do VRNN, SRNN and Clockwork VAE.
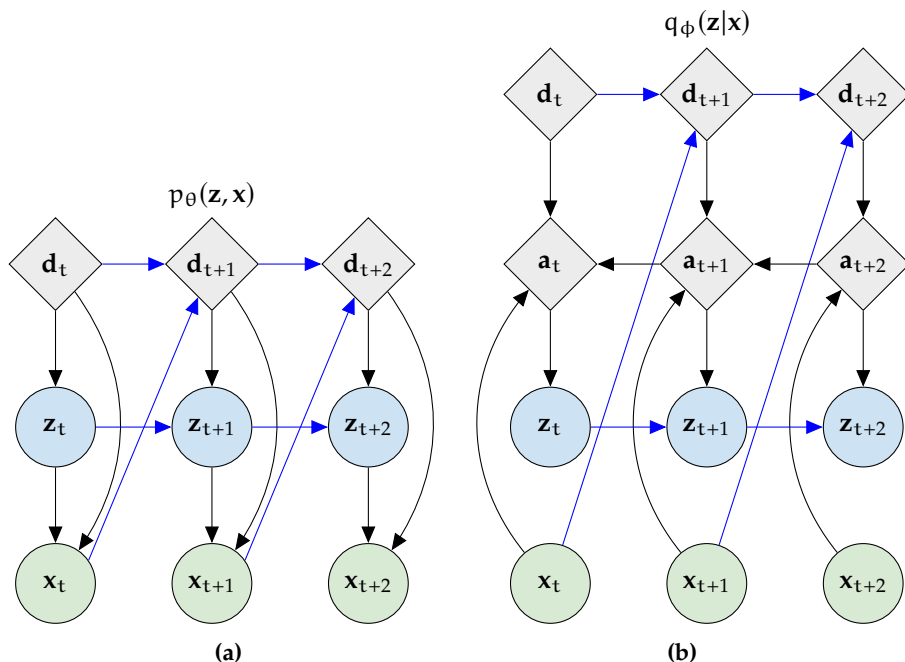
Figure C.5: SRNN [188] generative model $p(\mathbf{x}, \mathbf{z})$ in ((a)) and inference model $q(\mathbf{z}|\mathbf{x})$ in ((b)) unrolled over three steps in the observed variable. Blue arrows are shared between the inference and generative models.
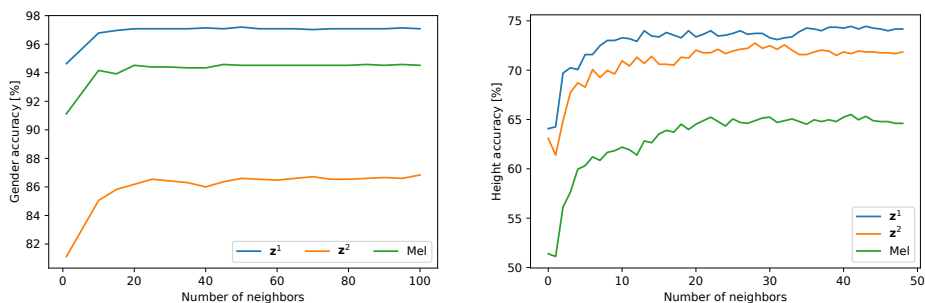
Figure C.6: Leave-one-out k-nearest-neighbor accuracy with different k for (a) the speaker's gender and (b) the height of male speakers (female speakers yield a similar result).

(a)   **(b)**
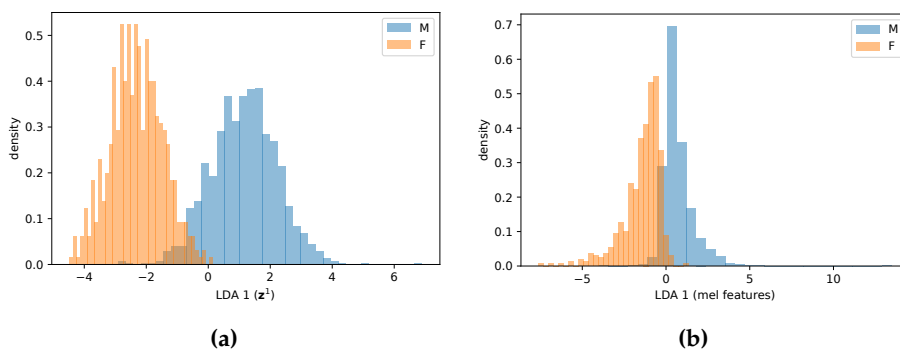
Figure C.7: Clustering of speaker gender in a one-dimensional linear subspace defined by a linear discriminant analysis of the CW-VAE latent space and of a time-averaged Mel spectrogram. The total overlap is slightly smaller in the subspace of the CW-VAE latent space and the separation between the distribution peaks is larger.

(a)   **(b)**
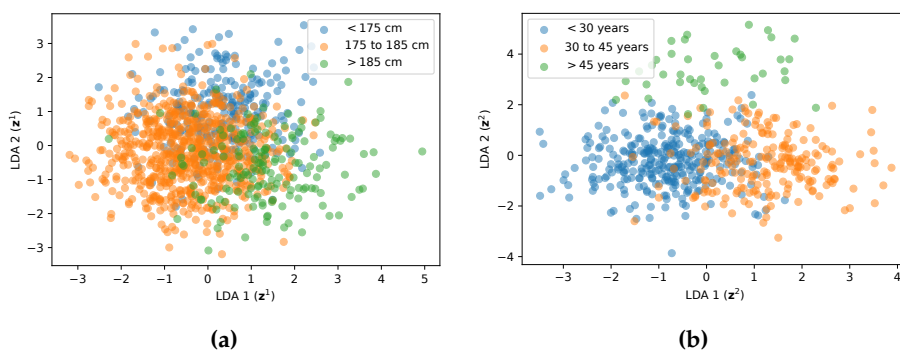
Figure C.8: (a) Clustering of speaker height for male speakers and (b) speaker age for female speakers in a two-dimensional linear subspace defined by a linear discriminant analysis of the CW-VAE latent space.
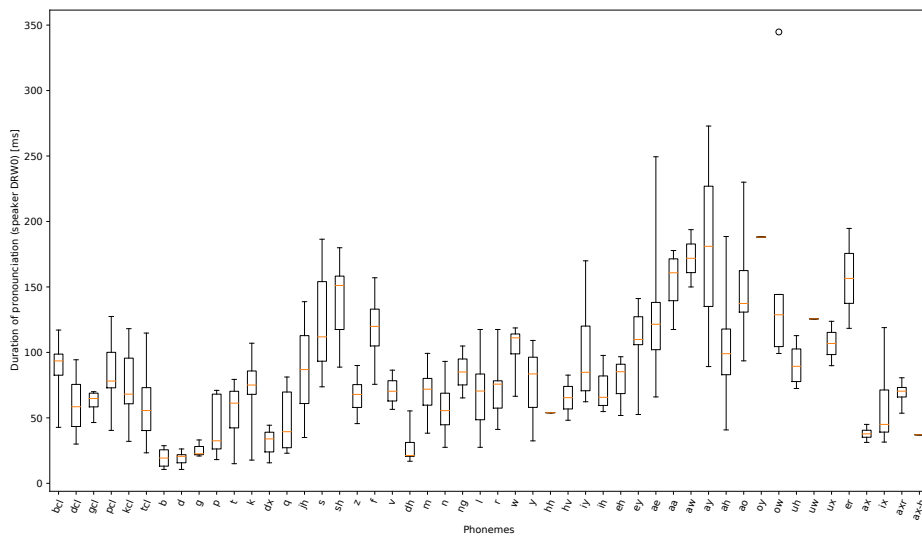
(a)
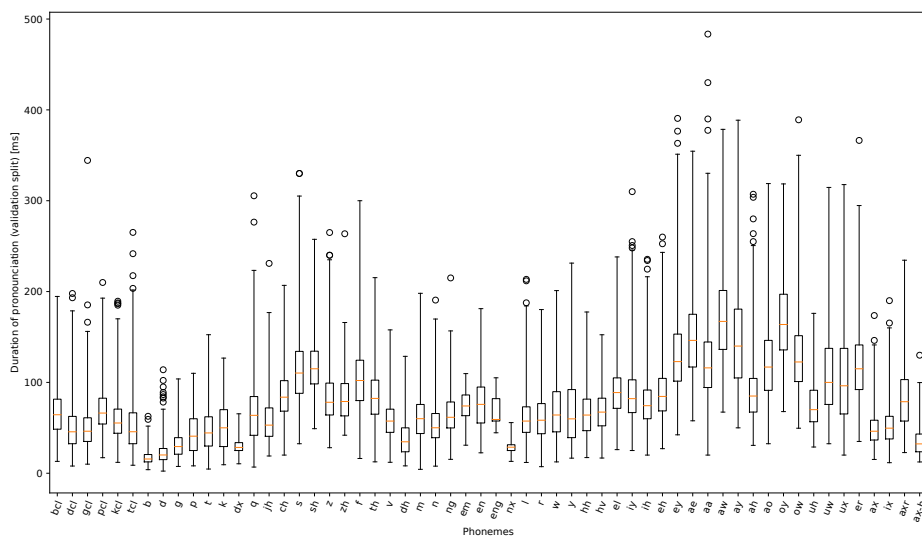
(b)

Figure C.9: Box plots of the duration of the pronunciation of phonemes in TIMIT
for a specific speaker DRW0 in ((a)) and globally in ((b)). Not all phonemes are
pronounced by speaker DRW0 over the course of their 10 test set sentences and
hence they are missing from the x-axis compared to the global durations.

Supplementary material: Model-Agnostic Out-of-Distribution Detection Using Combined Statistical Tests

D.1 Crude approximation of the Fisher information

The Fisher information is defined as:

$$I(\theta) = \mathbb{E}_{x \sim p_\theta}[\nabla \log p_\theta(x) \nabla \log p_\theta(x)^\top]. \tag{D.1}$$

A crude diagonal approximation can be computed by simply estimating the diagonal of $I(\theta)$ and setting all off-diagonal elements to zero. Such diagonal approximations have been used in machine learning for decades: for instance, Lecun and Soulie Fogelman [379, Section 3.12.2] used a similar approximation of the Hessian matrix, and called it "outrageously simplifying". Much more complex approximations have been derived, although diagonal approximations have been consistently used (e.g. by [348], who used essentially the same approximation in a supervised context), and are linked to several adaptive optimisation techniques like Adam [342] or RMSProp [638]. A good discussion on these issues is provided in Martens's (2020) recent review.

The approximation we used in the paper works as follows:

- By using the training examples $x_1, ..., x_T$, we form the estimate

$$D_T(\theta) = \frac{1}{T} \sum_{t=1}^{T} \text{diag}(\nabla \log p_\theta(x_t)^2),$$

  where the square in $\nabla \log p_\theta(x_t)^2$ is computed elementwise.

- While we could directly use $D_T(\theta)$ as an estimate. A slightly more refined approach is to slightly regularise $D_T(\theta)$. Following Martens [443], our final estimate of the Fisher information matrix is

$$\hat{I}_T(\theta) = (D_T(\theta) + \varepsilon)^\xi, \tag{D.2}$$

  with all operations performed elementwise. The diagonal matrix $\hat{I}_T(\theta)$ is then easy to invert and can be used to compute our statistics.

How to choose ε and ξ?   The Adam optimizer uses a similar estimate, with default hyperparameters $\varepsilon = 10^{-8}$ and $\xi = 1$. As argued by Martens [443], it can be interesting to use $\xi < 1$ in order to diminish the influence of extreme values of $D_T(\theta)$. In particular, Martens [443] suggests taking $\xi = 0.75$. When $\xi \longrightarrow 0$, then $\hat{I}_T(\theta)$ will approach the identity matrix. We tested the two settings by using a PixelCNN++ trained on CIFAR. Results are shown in table D.1. In terms of OOD detection, it seems that using $\varepsilon = 10^{-8}$ and $\xi = 1$ is slightly better. All results presented in the paper and in the supplementary material are computed by using $\varepsilon = 10^{-8}$ and $\xi = 1$.

A few notes on the computation of $D_T(\theta)$ While it seems more sensible to use samples $x_1, ..., x_m \sim p_\theta$ from the model, we decided to simply reuse the training data $x_1, ..., x_T$ instead. There are two computational advantages to this. The first one is that sampling many data points can be expensive (in particular for deep autoregressive models à la PixelCNN). The second advantage is that, if we wish to compute a MMD statistic, such as the MMD with the Fisher kernel or the MMD typicality (that require the average of gradient or the average log-likelihood over the training), computing the average of the square of the gradient costs very little. One can just do a single loop over the data, and use the usual formulas for online estimation of a mean, see algorithm 1.

Do we really need to approximate the diagonal of $I(\theta)$?   Another possibility is to just use the identity matrix as FIM instead of of approximating the diagonal through the procedure explained above. In our experiments (see table D.3 and table D.6), we can see that sometimes using the identity matrix seems to work equally well or a bit better for some models trained on FashionMNIST and CIFAR10. However, when we train on SVHN or MNIST, there are a cases where the statistic that is using the identity matrix as approximation fails, sometimes being worse than random chance. In those setting, using the diagonal approximation leads to way better results. Therefore, considering a test statistic that uses the diagonal approximation of the FIM is more robust for OOD detection.

D.2   THE MAHALANOBIS SCORE AS MMD

Lee et al. [389] introduced a simple metric to perform OOD detection with a trained deep classifier. The key idea is to train a simple generative model (linear discriminant analysis) in the feature space of the classifier. Let y denote the labels, and $z = f(x)$ the data in feature space. In the simplest case, f is just the trained deep net devoid of the last softmax layer. The linear discriminant analysis model is

$$y \sim \mathrm{Cat}(\pi), \quad z|y \sim \mathcal{N}(\mu_y, \Sigma), \tag{D.3}$$

Table D.1: AUROC↑ for single-sample OOD detection. Comparison between two different estimates of the Fisher information matrix. For (‡) we used the Adam parameter choice, i.e. $\varepsilon = 10^{-8}$ and $\xi = 1$. For (§), instead, we used $\varepsilon = 10^{-8}$ and $\xi = 0.75$, as suggested by Martens [443]. As a result, we have that using Adam parameters choice is slightly better for our task.

	CIFAR10 (IN) / SVHN (OUT)			
MODELS	MMD DIAGONAL	TYPICALITY	SCORE STAT	FISHER'S METHOD
PIXELCNN++ (model2) (‡)	0.7070	0.6498	0.7067	0.7300
PIXELCNN++ (model2) (§)	0.6881	0.6498	0.6878	0.7176

(‡) With $\varepsilon = 10^{-8}$ and $\xi = 1$
(§) With $\varepsilon = 10^{-8}$ and $\xi = 0.75$

where $\mu_1, ..., \mu_K$ are class-dependent means, Σ a common covariance matrix, and $\pi_1, ..., \pi_K$ are the class proportions, estimated by maximum-likelihood. The *Mahalanobis score* is then

$$M(x) = \max_{k \in \{1,...,K\}} -(z - \mu_k)^\mathsf{T} \Sigma^{-1} (z - \mu_k), \qquad (D.4)$$

which may be rewritten

$$M(x) = \max_{k \in \{1,...,K\}} p(z|k), \qquad (D.5)$$

under the assumption of equal class proportions (i.e. $\pi_1 = ... = \pi_K = 1/K$).

We show here that it is possible to re-interpret this score as a MMD score with a certain Fisher kernel. The generative model induced on z by linear disciminant analysis is a Gaussian mixture:

$$p_{\pi,\mu,\Sigma}(z) = \sum_{k=1}^{K} \pi_k \mathcal{N}(z|\mu_k, \Sigma). \qquad (D.6)$$

If we want a powerful deep kernel, it seems somewhat natural to consider the Fisher kernel associated with this generative model. The most important part of this mixture model are arguably the class-specific means (indeed, the model has been trained to discriminate the classes as well as possible). Therefore, we will only include these means in the Fisher kernel, and look at

$$\Phi_{\text{Fisher}}(x) = I(\mu)^{-1/2} \nabla_\mu \log p_{\pi,\mu,\Sigma}(z), \qquad (D.7)$$

assuming that π and Σ are fixed at their maximum likelihood estimates. Similar mixture-based Fisher kernels have been very popular in the past, and were actually a key element of state-of-the art classification models on Imagenet before

deep nets won the competition [525]. Our idea is to re-use ideas introduced by this computer vision litterature. Under the assumption that the Gaussian clusters are well-separated, Tanaka, Torii, and Okutomi [631], extending an earlier analysis of Sánchez et al. [576, Appendix A], showed that

$$[\Phi_{\text{Fisher}}(x)]_{\mu_k} \approx \sqrt{\frac{p(z|k)}{\pi_k}} \Sigma^{-1/2}(z - \mu_k). \tag{D.8}$$

Now, using the fact that the expected value of the score is approximatively zero, we can write that

$$\text{MMD}^2_{\Phi_{\text{Fisher}}} \approx \sum_{k=1}^{K} \| [\Phi_{\text{Fisher}}(x)]_{\mu_k} \|_2^2 \approx \sum_{k=1}^{K} \frac{p(z|k)}{\pi_k} (z - \mu_k)^\top \Sigma^{-1}(z - \mu_k). \tag{D.9}$$

Using again the fact that the clusters are well-separated, we may say that $z|k$ is approximatively a point mass at the most probable label, i.e. that $p(z|k) \approx \delta_k^{\text{argmax}_c p(z|c)}$. This leads to the approximation

$$\text{MMD}^2_{\Phi_{\text{Fisher}}} \approx \max_{k \in \{1,\dots,K\}} \frac{1}{\pi_k} (z - \mu_k)^\top \Sigma^{-1}(z - \mu_k). \tag{D.10}$$

Finally, assuming that the class proportions are equal leads to the equivalence of $\text{MMD}_{\Phi_{\text{Fisher}}}$ and the Mahalanobis score.

D.3   MORE INFORMATION ON THE EXPERIMENTAL SETUP

D.3.1   A BIT MORE BACKGROUND

The three considered DGMs are both parametrized by neural networks but they differ in the way they model the data distribution of interest. Assume we are interested in approximating a target distribution $p^*(\mathbf{x})$, for example a distribution of natural images, as it is done when using CIFAR10. PixelCNN++ is an autoregressive model and it models $p^*(\mathbf{x})$ as a product of conditional distribution over the variables, i.e. $p(\mathbf{x}) = p(x_1) \prod_{d=2}^{D} p(x_d \mid \mathbf{x}_{<d})$, where $\mathbf{x}_{<d} = [x_1, \dots, x_{d-1}]^\top$. Glow is a normalizing flow model and it approximate $p^*(\mathbf{x})$ by using a sequence of bijective transformations starting from a simple distribution, also called base distribution. If we use only a single invertible transformation f, the normalizing flow is defined as $\mathbf{x} = f(\mathbf{z})$, where $\mathbf{z} \sim p_Z(\mathbf{z})$, and $p_X(\mathbf{x}) = p_Z(\mathbf{z})|\det J_f(\mathbf{z})|^{-1}$, where we used the change of variable formula. For these two types of model we have a tractable likelihood that can be used to optimize the model parameters. The Variational Autoencoder (VAE), instead, is a framework to model the

data with a latent variable model, i.e. $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x} \mid \mathbf{z})p(\mathbf{z})$, where $\mathbf{x}$ is the observed input data and $\mathbf{z}$ is a stochastic latent variable and the prior distribution $p(\mathbf{z})$ is usually a standard Normal. Since the posterior $p(\mathbf{z} \mid \mathbf{x})$ is not tractable, a variational distribution $q_\phi(\mathbf{z} \mid \mathbf{x})$ is used as an approximation. Due to the intractability of the posterior, we cannot directly optimize the likelihood of the model, but instead the model parameters are optimized by maximizing the evidence lower bound (ELBO): $\log p_\theta(\mathbf{x}) \geqslant \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \frac{p(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] \equiv \mathcal{L}$. In this work we are considering an Hierarchical VAE (HVAE) with bottom-up inference as done in [245]. This is an extension of the VAE framework that consider an hierarchy of L latent variables $\mathbf{z} = \mathbf{z}_1, \ldots, \mathbf{z}_L$. The bottom-up inference is defined as $q_\phi(\mathbf{z} \mid \mathbf{x}) = q_\phi(\mathbf{z}_1 \mid \mathbf{x}) \prod_{i=2}^{L} q_\theta(\mathbf{z}_i \mid \mathbf{z}_{i-1})$, while the generative path is top-down, meaning $p_\theta(\mathbf{x} \mid \mathbf{z}) = p(\mathbf{x} \mid \mathbf{z}_1)p_\theta(\mathbf{z}_1 \mid \mathbf{z}_2) \cdots p_\theta(\mathbf{z}_{L-1} \mid \mathbf{z}_L)$. This is still trained by maximizing the ELBO. For a more in-depth explanation of these models we refer to their papers.

D.3.2   GENERATIVE MODEL DETAILS

We will briefly describe the different model architectures and training procedures used in this paper. Since most of the models are taken from public code repositories and related papers, we will mostly invite the reader to have a look at the cited paper for a more in-depth description of the training details. For MNIST, CIFAR10, and FashionMNIST we used 3000 examples from the test set as validation set. For SVHN, instead, we used 6032 datapoints from the test set as validation, leaving the remaining 20000 examples as test set. In table D.2, we reported test log-likelihood of the models used in this paper.

PixelCNN++   For PixelCNN++ we used the code available in this repository.[43] For the greyscale images, we used one residual block per stage with 32 filters and 5 logistic components in the discretized mixture of logistics. For natural images, instead, we used 5 residual blocks per stage with 160 filters and 10 components in the mixture. We trained all the models using Adam optimizer.

Glow   For training Glow models we follow Kirichenko, Izmailov, and Wilson [347] and their repository.[44] They closely follow Nalisnick et al. [472] and [339] implementation for multi-scale Glow, where a scale is defined as the sequence of actnorm, invertible 1×1 convolution and coupling layers. While Kirichenko, Izmailov, and Wilson [347] only considers the RMSProp optimizer, we trained two different models, one using RMSProp and one using Adam with batch-size

---

[43]https://github.com/pclucas14/pixel-cnn-pp
[44]https://github.com/PolinaKirichenko/flows_ood

Table D.2: Test log-likelihood (bits/dim) on MNIST, FashionMNIST, SVHN, and CIFAR10 achieved by the models used in the paper.

MODELS TRAINED ON FASHIONMNIST	
MODELS	LOG-LIKELIHOOD (BITS/DIM)
PixelCNN++ (dropout)	2.75
PixelCNN++ (no dropout)	2.72
Glow (RMSProp)	3.04
Glow (Adam)	3.02
HVAE (**)	0.43

MODELS TRAINED ON MNIST	
MODELS	LOG-LIKELIHOOD (BITS/DIM)
PixelCNN++ (dropout)	0.90
Glow (RMSProp)	1.32
Glow (Adam)	1.30
HVAE (**)	0.16

MODELS TRAINED ON CIFAR10	
MODELS	LOG-LIKELIHOOD (BITS/DIM)
PixelCNN++ (model1)	2.94
PixelCNN++ (model2)	2.94
Glow (RMSProp)	3.62
Glow (Adam)	3.62
HVAE	3.87

(**) Binarized FashionMNIST

MODELS TRAINED ON SVHN	
MODELS	LOG-LIKELIHOOD (BITS/DIM)
PixelCNN++ (dropout)	1.58
Glow (RMSProp)	2.23
Glow (Adam)	2.21
HVAE	2.38

(**) Binarized MNIST

32. For the greyscale dataset our Glow is made up of 2 scales with 16 coupling layers, and a 3-layers highway network with 200 hidden units is used to predict the scale and shift parameters. For CIFAR10 and SVHN, instead, we used 3 scales with 8 coupling layers, and 400 hidden units for the 3-layers highway network. For a more in-depth description, we refer to the codebase and the Appendix C of Kirichenko, Izmailov, and Wilson [347].

Hierarchical VAE We follow [245] for both model architecture design and training choices for our hierarchical VAEs. We used their open-sourced repository.[45] As mentioned in the paper, the HVAE model we used has a bottom-up inference path and a top-down generative path. We trained each model for 1000 epochs using Adam optimizer with learning rate $3e-4$ and a batch-size of 128. All models were initialized using the data-dependent initialization and they used weight-normalization [574]. In addition to that, we always consider a hierarchy of three latent variables. For greyscale images (MNIST and FashionMNIST) we used a latent dimension of $8-16-8$ for $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3$ respectively, while for natural images (CIFAR10 and SVHN) we used $8-16-32$. For a more in depth description of the model, we refer to Appendix B of Havtorn et al. [245].

---

[45] https://github.com/JakobHavtorn/hvae-oodd
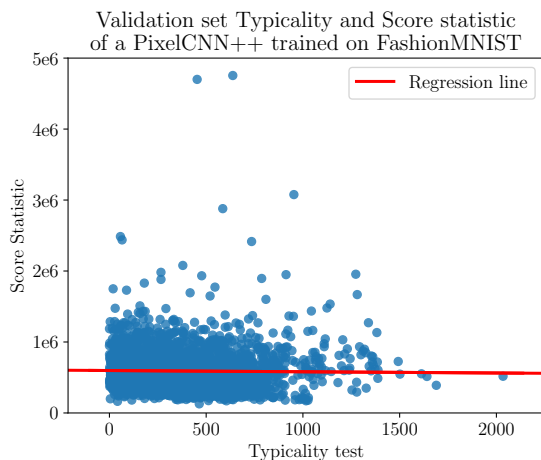
Figure D.1: Correlation of typicality test and score statistic computed on the validation set using a PixelCNN++ trained on FashionMNIST. The correlation coefficient is -0.014. This can also be seen by looking at the regression line, which is almost straight.

D.4   ADDITIONAL RESULTS

D.4.1   TYPICALITY TEST AND SCORE STATISTIC ARE UNCORRELATED

To test if the typicality test and the score statistic are uncorrelated, we plot the two scores computed on the validation set. As can be seen from figure D.1, we have that the two measures are not correlated as it is also highlight by the correlation coefficient.

D.4.2   HARMONIC MEAN

In the paper we mentioned that another way to combine p-values from different test statistics is the Harmonic mean [703]. This is defined as:

$$\mathring{p} = \frac{\sum_{i=1}^{k} w_i}{\sum_{i=1}^{k} w_i/p_i}, \tag{D.11}$$

where $w_1, \ldots, w_k$ are weights that sum up to 1. In our setting, we considered equal weights, i.e. $w_i = 1/k$. Therefore, if we simply consider two test statistics T_1 and T_2 and corresponding p-values p_1 and p_2, the harmonic mean p-values

becomes:

$$\mathring{p} = \frac{2p_1 p_2}{p_1 + p_2}. \tag{D.12}$$

As expected, this combination should work better when the statistics that we are combining are somewhat correlated. Indeed, since in our setting we have that the typicality and the score statistic are independent, we would expect this to work worse than the Fisher's combination. This is confirmed by table D.4, where we are reporting the results when combining the two statistics using the three different ways we analyzed.

D.4.3   RESULTS CONSIDERING MAXIMUM-MEAN-DISCREPANCY

In section 5.4, we discussed the relationship between the maximum-mean-discrepancy with a Fisher kernel and the score statistic and the gradient norm, which depends on the choice of approximation of the Fisher information matrix we use. In table D.3 we reported also the AUROC scores for the MMD with Fisher kernel considering both the diagonal approximation of the FIM (called *MMD diagonal* in the table) and the FIM being the identity matrix (called *MMD identity*). As expected, we have that the AUROC of the MMD with the diagonal approximated FIM is pretty close to the AUROC we obtained by using the score statistic. Likewise, we have that the AUROC of MMD with the identity matrix as FIM is close to the gradient norm when we trained on FashionMNIST and CIFAR10.

So, why did we decide to use the score statistic instead of the MMD with Fisher kernel and diagonal approximation of the FIM? The main reason is Occam's razor. If we have two things that work equally well, we should keep the simplest one. In our case, we have that for computing the MMD with the Fisher kernel, we need to compute both the average gradient and the FIM using the training set. For the score statistic, instead, we just need the FIM. In addition to that, from all our experiments (see table D.3 and table D.6) we do not have any evidence for one statistic working better than the other, because they are always pretty close to each other.

D.4.4   VARIABILITY WITHIN THE SAME MODEL IN DIFFERENT CHECKPOINTS

As mentioned in the paper, we noticed that all statistics depend on choices we made about our model and the training procedure, such as deciding between Adam or RMSProp, or between using dropout or not. In addition to that, we find out that they can differ also within the same model at different checkpoints that obtain almost the same log-likelihood. Here we consider two Glow models, one trained with Adam and one using RMSProp on CIFAR10. For both, we consider

Table D.3: AUROC↑ for single-sample OOD detection. In this table we consider all the different single statistics we mentioned in the paper with models trained on FashionMNIST and CIFAR10. One can notice that MMD diagonal is pretty close to the score statistic and the MMD identity is close to the gradient norm, as expected (see Section 4.1 in the paper). Complementary results for models trained on MNIST and SVHN are in table D.6.

| | FashionMNIST (in) / MNIST (out) | | | | | |
| | Single statistics | | | | | |
Models	$\log p(x)$	$\|\nabla \log p(x)\|_2$	MMD diagonal	MMD identity	Typicality	Score stat
PixelCNN++ (dropout)	0.0762	0.8709	0.8903	0.8690	0.8314	0.8822
PixelCNN++ (no dropout)	0.1048	0.9532	0.9393	0.9539	0.7575	0.9381
Glow (RMSProp)	0.1970	0.8904	0.9115	0.8986	0.4807	0.9114
Glow (Adam)	0.1223	0.7705	0.8540	0.7217	0.6987	0.8745
HVAE	0.0653	0.8714	0.9574	0.8726	0.8336	0.9578

| | CIFAR10 (in) / SVHN (out) | | | | | |
| | Single statistics | | | | | |
Models	$\log p(x)$	$\|\nabla \log p(x)\|_2$	MMD diagonal	MMD identity	Typicality	Score stat
PixelCNN++ (model1)	0.1553	0.8006	0.6406	0.8126	0.6457	0.6407
PixelCNN++ (model2)	0.1567	0.7923	0.7070	0.7955	0.6498	0.7067
Glow (RMSProp)	0.0630	0.8585	0.7929	0.8621	0.8651	0.7940
Glow (Adam)	0.0627	0.7844	0.7620	0.7838	0.8624	0.7655
HVAE	0.0455	0.8041	0.7268	0.7634	0.8845	0.7334

two checkpoints that achieve the same test log-likelihood. Those trained with Adam get a log-likelihood of 3.63 bits/dim, while the ones trained with RMSProp get 3.62 bits/dim. Results are shown in table D.5. It can be noticed, that although the models are similar in terms of test bits/dim the statistics vary a lot, mostly when training with RMSProp.

D.4.5 Benjamini-Hochberg procedure when training on CIFAR10

In the main paper we focused on the Benjamini-Hochberg procedure applied to a model trained on FashionMNIST. Although one should use a False Discovery Rate control procedure when the statistics we are using are strong, for completeness, we will present what happens when we apply the BH procedure on a model trained on CIFAR10. In figure D.2, we report the Type I error ratio and the Type II error ratio for different significance levels α. We can see that we can actually control the FDR for $\alpha > 0.2$, and for these significance levels we are actually controlling the FDR. What is happening for $\alpha < 0.2$? We have that the procedure is only rejecting 5 hypotheses and all these hypotheses corresponds to in-distribution examples. Therefore, we have that the ratio of Type I error is still low, but we are making a lot of Type II errors because we are accepting all

Table D.4: AUROC↑ for single-sample OOD detection. Comparison between the three methods we mentioned to combine different statistics. Since the typicality and the score statistic are not correlated, we have that the Fisher's method is mostly working better than the other two methods.

	FASHIONMNIST (IN) / MNIST (OUT)		
	COMBINATIONS		
MODELS	FISHER'S METHOD	HARMONIC MEAN	DOSE$_{\text{KDE}}$
PIXELCNN++ (dropout)	0.9369	0.9148	0.8822
PIXELCNN++ (no dropout)	0.9536	0.9392	0.9382
GLOW (RMSProp)	0.8598	0.8853	0.8901
GLOW (Adam)	0.8839	0.8632	0.8752
HVAE	0.9708	0.9569	0.9630

	CIFAR10 (IN) / SVHN (OUT)		
	COMBINATIONS		
MODELS	FISHER'S METHOD	HARMONIC MEAN	DOSE$_{\text{KDE}}$
PIXELCNN++ (model1)	0.6826	0.6667	0.6571
PIXELCNN++ (model2)	0.7300	0.7105	0.7243
GLOW (RMSProp)	0.8683	0.8551	0.8510
GLOW (Adam)	0.8613	0.8493	0.8588
HVAE	0.8699	0.8525	0.8245

Table D.5: AUROC↑ for single-sample OOD detection. In this table we are comparing two different Glow models trained on CIFAR10 by considering two different checkpoints with almost the same test log-likelihood. We can see that both statistics vary a bit.

	CIFAR10 (IN) / SVHN (OUT)			
	SINGLE STATISTICS		COMBINATION	
MODELS	TYPICALITY	SCORE STAT	FISHER'S METHOD	DOSE$_{\text{KDE}}$
GLOW (RMSProp) {*check1*}	0.8651	0.7940	0.8683	0.8510
GLOW (RMSProp) {*check2*}	0.8532	0.6894	0.8275	0.7815
GLOW (Adam) {*check1*}	0.8624	0.7655	0.8613	0.8588
GLOW (Adam) {*check2*}	0.8558	0.7327	0.8402	0.8303

the examples whose hypotheses should be rejected.

D.4.6 RESULTS WHEN TRAINING ON MNIST AND SVHN

We also evaluated our methods in the two dataset pairs, MNIST against Fashion-MNIST and SVHN against CIFAR10, that are usually considered easier than the tasks presented in the main paper. For both tasks, we trained two Glow models, one trained with Adam and one trained with RMSProp, one PixelCNN++
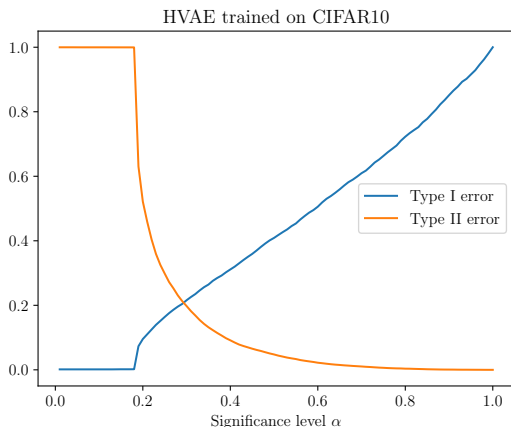
Figure D.2: Type I and Type II errors versus the significance level α on the combination values. We can control the FDR only for $\alpha > 0.2$ in this case. For $\alpha > 0.2$, since we are using Benjamini-Hochberg procedure, we get that the Type I error stays below identity line.

trained with dropout and a hierarchical VAE. Results are reported in table D.7. We can see that almost all the statistics we considered are able to almost perfectly distinguish between the in-distribution test-set and the OOD test-set. However, we can notice that the gradient norm is failing sometimes both when we trained on CIFAR10 and when we trained on FashionMNIST. From table D.6, instead, it is clear that we need to approximate the diagonal of the Fisher Information Matrix because if we simply consider the identity matrix, this will also fail as the gradient norm is doing.

D.4.7   APPLICATION OF OUR METHOD TO GAUSSIAN MIXTURE MODEL AND PROBABILISTIC PCA

Since the method we propose is model-agnostic, we show that it can be used for out-of-distribution detection also using two simple generative models, Gaussian Mixture Model (GMM) and Probabilistic PCA (PPCA). We consider the two pairs of datasets as before, i.e. FashionMNIST vs MNIST and CIFAR10 vs SVHN. Results can be seen in table D.8 and table D.9. For both GMM and PPCA trained on FashionMNIST the likelihood can be used to perform OOD detection. Indeed, in this setting, they are not assigning higher likelihood to OOD data as it is the case for DGMs. This happens instead when we fit these models on CIFAR10. How-

Table D.6: AUROC↑ for single-sample OOD detection. In this table we consider all the different single statistics we mentioned in the paper with models trained on MNIST and SVHN. In this case, it is important to notice that the gradient norm and the MMD identity sometimes fail to a different extent. Complementary results for models trained on MNIST and SVHN are in table D.3

	MNIST (in) / FashionMNIST (out)					
	SINGLE STATISTICS					
MODELS	$\log p(x)$	$\|\nabla \log p(x)\|_2$	MMD DIAGONAL	MMD IDENTITY	TYPICALITY	SCORE STAT
PixelCNN++ (dropout) (†)	0.9999	0.8534	0.9993	0.8608	0.9996	0.9993
Glow (RMSProp)	0.9997	0.9936	0.9942	0.6609	0.9991	0.9936
Glow (Adam)	0.9999	0.6506	0.9993	0.9124	0.9997	0.9992
HVAE	0.9999	0.9998	0.9999	0.9999	0.9999	0.9999

	SVHN (in) / CIFAR10 (out)					
	SINGLE STATISTICS					
MODELS	$\log p(x)$	$\|\nabla \log p(x)\|_2$	MMD DIAGONAL	MMD IDENTITY	TYPICALITY	SCORE STAT
PixelCNN++ (dropout)	0.9820	0.2670	0.9543	0.3185	0.9590	0.9543
Glow (RMSProp)	0.9917	0.9180	0.9824	0.9317	0.9830	0.9823
Glow (Adam)	0.9913	0.5658	0.9653	0.7096	0.9779	0.9641
HVAE	0.9943	0.1011	0.9865	0.4508	0.9857	0.9862

(†) Trained using 50000 datapoints

ever, this behaviour can be due to the fact that they are really poor generative models for this dataset. It is also surprising that when training on CIFAR10 the score statistic is failing in both models. We think that this is also due to the fact that both the GMM and the PPCA are far from being good generative models for this dataset.

D.4.8   More in depth analysis of the variability of the results for different HVAE

As we have pointed out before, test statistics and consequentially out-of-distribution performances can vary between the same model trained several times on the same dataset. To test the variability of the results shown in the main paper, we trained five different hierarchical VAEs and compute mean and standard deviations of the final AUROC scores. All models have the same architecture and were trained with the same procedure. Results can be found in table D.10. For the models trained on CIFAR10, most of the variability in terms of performance is due to the score statistic, which has the highest standard deviation. When training on FashionMNIST, instead, it seems that the typicality performance is the one varying the most between the five models.

Table D.7: AUROC↑ for single-sample OOD detection when training on MNIST and testing against FashionMNIST and when training on SVHN and testing against CIFAR10. As before, Fisher's method is the combination of the typicality test and the test statistic. These are also combined using DoSE. Complementary results are in table D.4.

| | MNIST (IN) / FashionMNIST (OUT) | | | | | |
| | SINGLE STATISTICS | | | | COMBINATION | |
MODELS	$\log p(x)$	$\|\nabla \log p(x)\|_2$	TYPICALITY	SCORE STAT	FISHER'S METHOD	DoSE$_{KDE}$
PixelCNN++ (dropout) (†)	0.9999	0.8534	0.9996	0.9993	0.9999	0.9999
Glow (RMSProp)	0.9997	0.9936	0.9991	0.9936	0.9992	0.9994
Glow (Adam)	0.9999	0.6506	0.9995	0.9992	0.9998	0.9999
HVAE	0.9999	0.9998	0.9997	0.9999	0.9999	0.9999

| | SVHN (IN) / CIFAR10 (OUT) | | | | | |
| | SINGLE STATISTICS | | | | COMBINATION | |
MODELS	$\log p(x)$	$\|\nabla \log p(x)\|_2$	TYPICALITY	SCORE STAT	FISHER'S METHOD	DoSE$_{KDE}$
PixelCNN++ (dropout)	0.9820	0.2670	0.9590	0.9543	0.9914	0.9824
Glow (RMSProp)	0.9917	0.9180	0.9830	0.9823	0.9913	0.9913
Glow (Adam)	0.9913	0.5658	0.9779	0.9641	0.9883	0.9863
HVAE	0.9943	0.1011	0.9857	0.9862	0.9934	0.9862

(†) Trained using 50000 datapoints

Table D.8: AUROC↑ for single-sample OOD detection using a Gaussian mixture model (GMM). For Fisher's method we mean the combination of the typicality test and the test statistic. These are also combined using DoSE.

| | FashionMNIST (IN) / MNIST (OUT) | | | | | |
| | SINGLE STATISTICS | | | | COMBINATION | |
COMPONENTS	$\log p(x)$	$\|\nabla \log p(x)\|_2$	TYPICALITY	SCORE STAT	FISHER'S METHOD	DoSE$_{KDE}$
50	0.6627	0.5514	0.5196	0.8777	0.7689	0.8152
100	0.6872	0.5509	0.5575	0.8742	0.7965	0.7989

| | CIFAR10 (IN) / SVHN (OUT) | | | | | |
| | SINGLE STATISTICS | | | | COMBINATION | |
COMPONENTS	$\log p(x)$	$\|\nabla \log p(x)\|_2$	TYPICALITY	SCORE STAT	FISHER'S METHOD	DoSE$_{KDE}$
50	0.2335	0.6087	0.6759	0.3512	0.6098	0.6569
100	0.2372	0.6136	0.6714	0.3294	0.5898	0.6573

D.5 Yes, we should talk about CelebA

Out-of-distribution detection performance is not only influenced by the model architecture or the training process. Indeed, transformations applied to the input data play an important role by transforming a difficult task into an easier problem where the likelihood can detect OOD data. By looking at the different results for Glow trained on CIFAR10 and tested on CelebA shown in [254], [347], [467],

Table D.9: AUROC↑ for single-sample OOD detection using a Probabilistic PCA. For Fisher's method we mean the combination of the typicality test and the test statistic. These are also combined using DoSE.

	FASHIONMNIST (IN) / MNIST (OUT)					
	SINGLE STATISTICS				COMBINATION	
COMPONENTS	$\log p(x)$	$\|\nabla \log p(x)\|_2$	TYPICALITY	SCORE STAT	FISHER'S METHOD	DoSE$_{KDE}$
50	0.9727	0.9637	0.9587	0.9505	0.9635	0.9610
100	0.9557	0.9715	0.9309	0.9626	0.9566	0.9585

	CIFAR10 (IN) / SVHN (OUT)					
	SINGLE STATISTICS				COMBINATION	
COMPONENTS	$\log p(x)$	$\|\nabla \log p(x)\|_2$	TYPICALITY	SCORE STAT	FISHER'S METHOD	DoSE$_{KDE}$
50	0.0770	0.1494	0.8468	0.1308	0.7568	0.8210
100	0.0357	0.0778	0.8944	0.0755	0.7966	0.8830

Table D.10: Mean and standard deviation of the performance in terms of AUROC of our method. Quantities are computed by taking the performance of five different trained HVAEs both trained on CIFAR10 and FashionMNIST.

D_{OUT}	$\log p(x)$	TYPICALITY	SCORE STAT	FISHER'S METHOD	DoSE$_{KDE}$
	HVAE TRAINED ON CIFAR10				
SVHN	0.0631 (0.0008)	0.8711 (0.0028)	0.7808 (0.0255)	0.8844 (0.0140)	0.8519 (0.0194)
CIFAR100	0.5349 (0.0007)	0.5496 (0.0003)	0.5857 (0.0042)	0.5924 (0.0029)	0.5985 (0.0028)
CELEBA	0.9004 (0.0035)	0.8203 (0.0046)	0.7565 (0.0369)	0.8505 (0.0138)	0.8247 (0.0228)
	HVAE TRAINED ON FASHIONMNIST				
MNIST	0.2487 (0.0152)	0.5064 (0.0245)	0.9532 (0.0084)	0.9220 (0.01491)	0.9377 (0.0126)

and [2] we can see that the AUROC scores obtain by the plain log-likelihood are pretty different. In [254] and [347] the log-likelihood gets a poor performance, confirming that CIFAR10-CelebA is a challenging pair for DGMs, while in [467] the likelihood is able to distinguish OOD data. While the main reason for these different results can be due to model implementation and training procedure, we decided to investigate how different transformations can influence OOD detection. Indeed, CelebA examples originally have a shape of $(218, 178, 3)$ and to transform them into $(32, 32, 3)$-shaped images, as CIFAR10, we have to resize them and then crop their center. The resize function is performing an interpolation, therefore we analyze how different interpolation strategies influence the OOD task.

We considered three different interpolations: bilinear (default in PyTorch), Lanczos, and nearest. As can be seen from figure D.3, these transformations mostly affect the sharpness of the images. In table D.11 we show how the OOD

Figure D.3: Comparison of different interpolation methods for CelebA dataset.

performance changes for our considered models when testing on CelebA where we applied different interpolations. We can notice that when using the bilinear interpolation we get results that are pretty similar to [254], [347], and [2] in terms of likelihood OOD performance. When using the nearest interpolation, instead, we get results that are closer to [467].

In conclusion, with these experiments, we wanted to highlight the importance of reporting the preprocessing steps used in loading CelebA in order to be able to make a fair comparison with the other proposed methods in the literature.

D.6 Comparison with the original DoSE statistics

As the last experiment, we study how our proposed method with our model agnostic statistic performs against DoSE using the original statistics proposed in [467]. For the VAEs model, they suggested to use the following 5 statistics: the posterior/prior cross-entropy $H[q_\phi(\mathbf{z} \mid \mathbf{x}), p(\mathbf{z})]$, the posterior entropy $H[q_\phi(\mathbf{z} \mid \mathbf{x})]$, the posterior/prior KL-divergence $D_{KL}[q_\phi(\mathbf{z} \mid \mathbf{x}) \mid\mid p(\mathbf{z})]$, the posterior expected log-likelihood $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log q_\phi(\mathbf{z} \mid \mathbf{x})]$, and the log-likelihood $\log \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right]$. For DoSE on Glow, instead, they considered three metrics: the log-likelihood $p_X(\mathbf{x} \mid \theta_n)$ and its two components, i.e. the log-probability of the latent variable $p_Z(\mathbf{z} \mid \mathbf{x}, \theta_n)$ and the log-determinant of the Jacobian $\log |J_f(\mathbf{x})|$.

In this setting, since DoSE is using statistics that are HVAE and Glow specific, it is not model agnostic anymore. Indeed, we cannot use those statistics also for a PixelCNN++ for example or any other DGM. We want also to highlight that the models used in [467] are a bit different from the ones used in this work. For example, they are considering a beta-VAE with only one stochastic layer, while in our case we used a HVAE with 3-stochastic layers.

D.7 Algorithmic implementation

A pseudocode describing step-by-step how to implement our method is given in algorithm 1.

Table D.11: AUROC↑ for single-sample OOD detection training on CIFAR10 and testing on CelebA considering all the three interpolations for CelebA.

| | CIFAR10 (IN) / CELEBA (OUT) (†) | | | | | |
| | SINGLE STATISTICS | | | | COMBINATION | |
MODELS	$\log p(x)$	$\|\nabla \log p(x)\|_2$	TYPICALITY	SCORE STAT	FISHER'S METHOD	DoSE$_{KDE}$
PixelCNN++ (model1)	0.7027	0.5856	0.5581	0.7001	0.6450	0.6931
PixelCNN++ (model2)	0.7034	0.4298	0.5554	0.7505	0.6879	0.7430
Glow (RMSProp)	0.5337	0.5616	0.3926	0.6561	0.5400	0.5866
Glow (Adam)	0.5308	0.5820	0.3914	0.5850	0.4818	0.5212
HVAE	0.5643	0.5214	0.4011	0.6712	0.5483	0.5987

| | CIFAR10 (IN) / CELEBA (OUT) (‖) | | | | | |
| | SINGLE STATISTICS | | | | COMBINATION | |
MODELS	$\log p(x)$	$\|\nabla \log p(x)\|_2$	TYPICALITY	SCORE STAT	FISHER'S METHOD	DoSE$_{KDE}$
PixelCNN++ (model1)	0.8284	0.5035	0.7399	0.6714	0.7477	0.7123
PixelCNN++ (model2)	0.8284	0.3530	0.7370	0.70088	0.7631	0.7446
Glow (RMSProp)	0.7556	0.4427	0.6222	0.7865	0.7423	0.7632
Glow (Adam)	0.7499	0.4800	0.6177	0.6442	0.6460	0.6467
HVAE	0.7561	0.4097	0.6051	0.6779	0.6775	0.6772

| | CIFAR10 (IN) / CELEBA (OUT) (‡) | | | | | |
| | SINGLE STATISTICS | | | | COMBINATION | |
MODELS	$\log p(x)$	$\|\nabla \log p(x)\|_2$	TYPICALITY	SCORE STAT	FISHER'S METHOD	DoSE$_{KDE}$
PixelCNN++ (model1)	0.9270	0.4196	0.8902	0.8320	0.9287	0.8908
PixelCNN++ (model2)	0.9270	0.3065	0.8886	0.8448	0.9339	0.9236
Glow (RMSProp)	0.9364	0.5345	0.8880	0.9286	0.9390	0.9423
Glow (Adam)	0.9322	0.5957	0.8829	0.8350	0.9017	0.8933
HVAE	0.8964	0.3515	0.8158	0.7952	0.8620	0.8455

(†) Bilinear interpolation
(‖) Lanczos interpolation
(‡) Nearest interpolation

Table D.12: Comparison between our method and DoSE using the original statistics. In these experiments we considered only Glow trained with Adam.

D_{OUT}	OUR METHOD	DoSE$_{orig}$
	GLOW TRAINED ON CIFAR10	
SVHN	**0.8613**	0.7819
CIFAR100	**0.5775**	0.5700
CELEBA	0.9017	**0.9663**
	GLOW TRAINED ON FashionMNIST	
MNIST	0.8839	**0.9568**
	HVAE TRAINED ON FashionMNIST	
MNIST	0.9383	**0.9762**
	HVAE TRAINED ON CIFAR10	
SVHN	0.8605	**0.8823**
CIFAR100	**0.5888**	0.5608
CELEBA	**0.8620**	0.8203

Algorithm 1 Computing p-values for OOD detection using a trained generative model.

Input: Training data $\mathbf{X} = (x_1, \ldots, x_m)^\top$, validation data $\mathbf{X}'$, trained model $p_\theta(x)$.

Approximation of the diagonal of the Fisher Information Matrix $I(\theta)$ *and average log-likelihood* $(1/m) \log p_\theta(x_1, \ldots, x_m)$, *indicated by* $L(\theta)$. *We do it in an online fashion.*
Initialize $I(\theta) = 0$ and $L(\theta) = 0$
For all $i \in \{1, \ldots, m\}$:
    Compute $\log p_\theta(x_i)$
    Compute $\nabla_\theta \log p(x_i \mid \theta)$
    Set $I(\theta) = \frac{1}{i+1} \cdot (i \cdot I(\theta) + (\nabla_\theta \log p_\theta(x_i))^2)$
    Set $L(\theta) = \frac{1}{i+1} \cdot (i \cdot L(\theta) + \log p_\theta(x_i))$

Estimation of distributions over the test statistics
Sample S M'-sized datasets from $\mathbf{X}'$ using bootstrap resampling.
(For single-sample OOD we just cycle through each example, see section 5.3)
Initialize $T^{\text{typicality}} = [\,]$ and $T^{\text{score}} = [\,]$
For every bootstrapped dataset $\mathbf{X}'_s = (x_1, \ldots, x_{M'})^\top$:
    Compute $\frac{1}{m'} \sum_{m'=1}^{M'} \log p_\theta(x_{m'})$
    Compute $\frac{1}{m'} \sum_{m'=1}^{M'} \nabla_\theta \log p_\theta(x_{m'})$
    Compute MMD Typicality for $x_{m'}$ by $\left\| \frac{1}{m'} \sum_{m'=1}^{M'} \log p_\theta(x_{m'}) - L(\theta) \right\|_2$ and
add it to $T^{\text{typicality}}$
    Compute Score statistic for $x_{m'}$ by $\left\| I(\theta)^{-1/2} \frac{1}{m'} \sum_{m'=1}^{M'} \nabla \log p_\theta(x_{m'}) \right\|_2$ and
add it to T^{score}
Return Two vectors of size S containing the two statistics for $T^{\text{typicality}}$ and T^{score}

Compute $\hat{F}^{\text{typicality}}$ and $\hat{F}^{\text{score}}$, the two empirical CDFs, from $T^{\text{typicality}}$ and T^{score}. For example, we used `statsmodels` library [587].

Given a test set $\tilde{x}_1, \ldots, \tilde{x}_n$:
(n = 1 corresponds to perform single-sample OOD detection)
    Compute $\frac{1}{n} \sum_{i=1}^{n} \log p_\theta(\tilde{x}_i)$ and $\frac{1}{n} \sum_{i=1}^{n} \nabla_\theta \log p_\theta(\tilde{x}_i)$
    Compute MMD Typicality $\tilde{t}$ and Score statistic $\tilde{s}$
    Compute p-values $p_T = 1 - \hat{F}^{\text{typicality}}(\tilde{t})$ and $p_S = 1 - \hat{F}^{\text{score}}(\tilde{s})$
    Combine the two p-values using Fisher's method (5.5)

SUPPLEMENTARY MATERIAL FOR: A RETROSPECTIVE STUDY ON MACHINE LEARNING-ASSISTED STROKE RECOGNITION FOR MEDICAL HELPLINE CALLS

E.1   DATA FLOW DIGRAM

In figure E.1 we present an overview of the data flow from the initial data sources to the final stroke dataset.

E.2   MACHINE LEARNING PIPELINE

E.2.1   MODEL TRAINING

We used stochastic gradient descent on mini-batches of data to train the stroke classification model. We used the Adam (adaptive moment estimation) optimisation algorithm and ensured an equal number of stroke positives and negatives in each batch by stratifying the class labels during sampling. We saved the model parameters after each epoch if the maximum F1-score (across all possible thresholds) improved in the validation dataset. We used the latest saved parameters as the final result of the run.

E.2.2   HYPERPARAMETERS

The selection of hyperparameters followed a simple two-stage process using validation data (table 1). First, a manual search was conducted by running different model configurations with varying numbers of epochs, updates per epoch, batch sizes, vectoriser types, and hyperparameters. Subsequently, a structured grid search was performed to further tune a subset of these hyperparameters.

Bag-of-words selection   Each transcript was transformed into a fixed-size bag-of-words vector to serve as input for the classification model. These vectors encode the occurrence of words and character n-grams within a fixed vocabulary. We selected vocabulary by first computing the χ^2-statistics for all word uni- and bi-grams and character three-, four-, and five-grams that occurred in more than ten training calls. We then retained the M highest-scoring word n-grams and M highest-scoring character n-grams, yielding 2M input features, where M represents a tuned hyperparameter. By complementing word n-grams with character n-grams, the model can use out-of-vocabulary words not included in the word
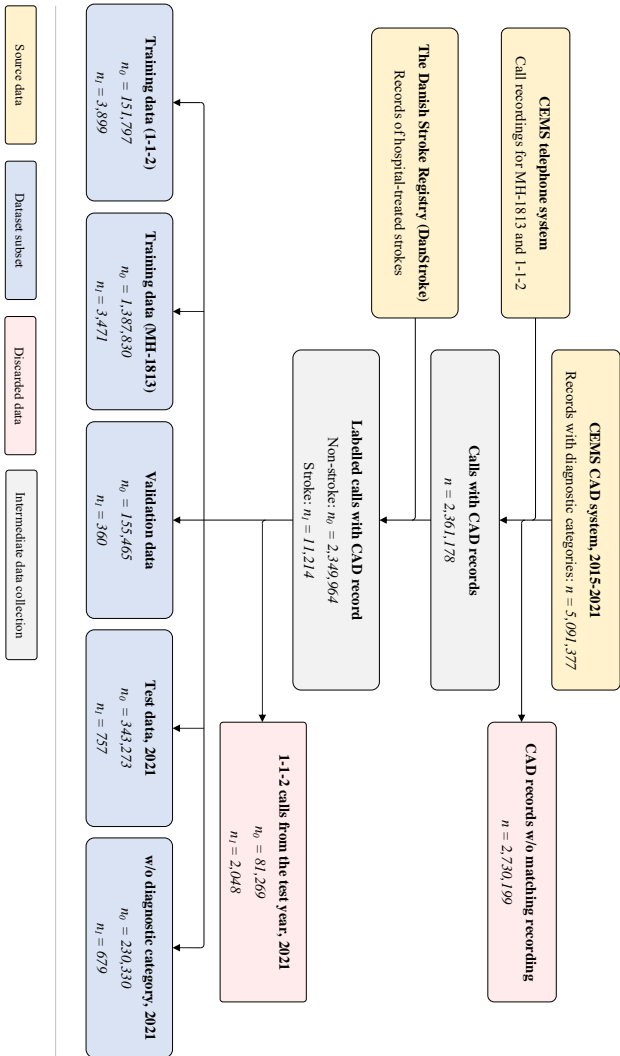
Figure E.1: Overview of data flow from the initial data sources to the final stroke dataset.

Table E.1: Overview of hyperparameters for training text classification models.

Name	Chosen value	Grid search range
Determined from initial heuristic, manual hyperparameter search on validation fold		
Epochs	30	-
Parameter updates per epoch	500	-
Batch size	128	-
Batch sampling	Label stratified (balanced)	-
Type of text vectoriser	Count vectorisation	-
Size of bag-of-words vector	10,000	-
Optimisation algorithm	Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$)	-
Learning rate schedule	Cosine annealing from start	-
Determined from grid search on validation fold		
Learning rate start	0.0003	{0.003, 0.0003}
Learning rate end	0.0003	{0.0003, 0.00003, 0.000003}
Model input dropout	0.50	{0.25, 0.50}
Model dropout	0.25	{0.25, 0.50}
Model configuration	[256, 128, 64, 32, 16]	{[64, 32, 16], [256, 128, 64, 32, 16]}

n-grams and robustly represent words misspelt by the speech recogniser. The feature vector was input into the classification model (figure E.2).

As part of our manual hyperparameter search, we trained the models using vectorisers of different sizes. We discovered that using 5,000-word n-grams and 5,000-character n-grams (M=5,000) struck a good trade-off between size, feature quality, and model performance, yielding 10,000 bag-of-words features.

E.2.3 Ensembling details

A common way to combine individual classification models into an ensemble is to use a voting scheme, such as majority voting, where a combined prediction is made based on the consensus among the individual models. However, this approach makes the ensemble not have a continuous output score. This is problematic for two reasons.

The lack of a continuous output score prevents the evaluation of the model's performance across a continuous range of thresholds required for plotting the receiver operating characteristic and precision-recall curves.

The lack of a continuous output score deteriorates the quality of assessing the effect of different words on ensemble performances (see section 9.4.5).

Therefore, we used a different ensemble method, which is briefly described in the main text. Herein, we provide a mathematically rigorous definition of the proposed method.

Let $z^{(n,d)}$ be the logit output of model n for transcript d, $t^{(n)}$ be the tuned logit threshold of model n, and N be the number of models in the ensemble. The

A. Schematic Overview of Stroke Classification Pipeline

B. Speech Recognition Model

C. Text Classification Model
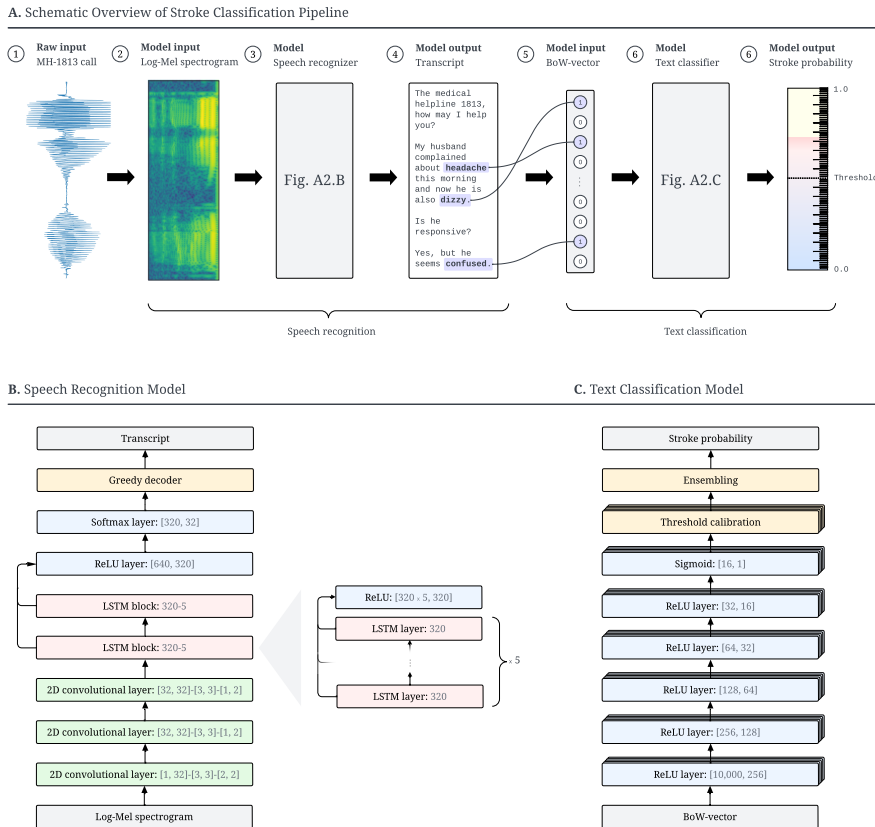
Figure E.2: Overview of machine learning pipeline. Panel A presents a schematic overview of the machine learning pipeline. The individual models are broken down in panels B and C. The 2D convolutional layers have parameters [*input channels*, *output channels*]-[*kernel width*, *kernel height*]-[*stride width*, *stride height*]. The long short-term memory (LSTM) blocks have parameters hidden units-number of layers. The rectified linear unit (ReLU), sigmoid, and softmax layers have parameters [input features, output features]. Joining arrows indicate the concatenation of two vector sequences along the feature dimension. The full set of model hyperparameters is listed in appendix E.2.2.

output score $p^{(n)}$ is then given by

$$p^{(d)} = \frac{1}{N} \sum_{n=1}^{N} \sigma \left(z^{(n,d)} - t^{(n)} \right) \ , \tag{E.1}$$

where $\sigma(\cdot)$ is the sigmoid function (or standard logistic function). The final ensemble prediction $s^{(d)}$ is then simply

$$s^{(d)} = I_{p^{(d)}>0.5} \left(p^{(d)} \right) \ , \tag{E.2}$$

where $I.(\cdot)$ is the indicator function that returns 1 if the subscript condition is satisfied and 0 otherwise.

E.3 Significance testing and confidence intervals

We used standard methods for significance testing and computing the confidence intervals. We used approximate methods owing to data size and to maintain computational feasibility [167, 170].

We performed *paired approximate permutation* tests by pairing each observation from the first sample to a random observation from the other sample (without replacement), while keeping each observation within its original sample. This allowed us to test the significance of the observed pairings on the chosen statistics, i.e. whether a significant difference was observed in the test statistics depending on whether the call-taker or model made the prediction. We used this approach to test whether

- the model performed better than the call-takers on the 2021 test set,

- including 112 training data improved the model's performance on the 2021 test set.

We performed *independent approximate permutation* tests by randomly assigning observations to either of the two samples (without replacement) while maintaining any differences in sample size. This approach allowed us to test the significance of the observed sample assignments, i.e. whether there was a significant difference in test statistics depending on whether the predictions were assigned to the model or call-taker. We used this approach to test whether.

- the model performed better on the 2021 test set with diagnostic categories than on the test set without diagnostic categories,

- the model performed better on men than women on the 2021 test set,

- the model performed better on the 65+ group than on the 18-64 group on the 2021 test set,

- the call-taker performed better on men than women on the 2021 test set,

- the call-taker performed better on the 65+ group than on the 18-64 group on the 2021 test set.

The p-values were not exact because we used approximate permutation tests. However, owing to the large dataset size and substantial number of observations, the estimated p-values had tight confidence intervals. We reported the upper bound of the 99% confidence interval on the p-value computed as the usual binomial distribution confidence interval.

$$\text{CI}(\text{p}) = \hat{p} \pm z\sqrt{\frac{\hat{p}(1 - \hat{p})}{N}} \ , \tag{E.3}$$

where N is the number of resamplings.

We computed *bootstrapped confidence intervals* for the statistics by resampling (with replacement) the predictions made by model or call-taker on a relevant subgroup and recomputing the relevant statistics for each bootstrap sample. This process established a bootstrap distribution of the statistic that was then used to estimate the standard error and compute the confidence intervals. We computed confidence intervals using the bootstrap distribution percentiles. This method yielded reliable results because our dataset was large, and the bootstrap distribution was symmetrical and centred on the observed statistic. No observed bootstrap distributions differed significantly from normal distributions (according to Anderson-Darling and Shapiro-Wilk tests). Therefore, confidence intervals computed alternatively as studentised bootstrap intervals (t-intervals) did not differ substantially from percentile confidence intervals. This further validated our tests.

We used $N = 15,000$ resamplings for permutation tests and $N = 8,192$ for confidence intervals.

E.4   SOFTWARE

We used Python version 3.8.10. PyTorch version 1.12.1 + cu113 was used to train the neural network models. We used SciKit-Learn (version 1.2.2) to perform bag-of-words vectorisation. We used NumPy version 1.23.5, Pandas version 1.5.3, Matplotlib version 3.7.1, and SciPy version 1.10.1 to perform data analysis, plotting, and testing.

E.5   Additional results: Model performance across demographics

When 1-1-2 data were not used for training (table E.2), the model performed significantly better in men than in women on the test set in terms of all metrics (p < 0.0001, paired approximate permutation test) and significantly better on the 65+ group than on the 18-64 group in terms of all metrics (p < 0.0001, paired approximate permutation test). As noted in the main text, both these statements were also true when 1-1-2 data were used for the training (table 9.2) and that including 1-1-2 data significantly improved overall performance (p < 0.0001, paired approximate permutation test). Compared to the test set (table 9.2), model performance on the 2021 calls without diagnostic category was significantly worse in all demographic subgroups (table E.3) in terms of all metrics (p < 0.0001, independent approximate permutation test), except for positive predictive value (PPV) for male (where p = 0.0056 [significant], independent approximate permutation test) and for the false positive rate (FPR) on females and individuals aged 65+ years (where p = 0.213 and p = 0.362, respectively [insignificant] , independent approximate permutation test).

Table E.2: Model performance grouped by sex and age [mean (95% CI)] when 1-1-2 training data is not used for training. NPV: negative predictive value, PPV: positive predictive value, FOR: false omission rate, CI: confidence interval.

	Model w/o 1-1-2 training data			
	Female	Male	18-64 years	65+ years
F1-score [%] ↑	28.7 (27.8-29.6)	36.2 (35.3-37.2)	20.5 (19.5-21.5)	39.4 (38.6-40.3)
Sensitivity [%] ↑	58.2 (56.7-59.9)	62.1 (60.7-63.6)	53.0 (50.9-55.0)	63.0 (61.9-64.3)
PPV [%] ↑	19.0 (18.3-19.7)	25.6 (24.8-26.4)	12.7 (12.0-13.4)	28.7 (27.9-29.5)
FOR [%] ↓ (1 - NPV)	0.077 (0.073-0.080)	0.102 (0.097-0.106)	0.034 (0.032-0.036)	0.318 (0.305-0.331)
FPR [%] ↓ (1 - specificity)	0.45 (0.446-0.463)	0.483 (0.473-0.494)	0.264 (0.258-0.270)	1.335 (1.308-1.362)

Table E.3: Model performance grouped by sex and age [mean (95% CI)] on the 2021 data without diagnostic category. NPV: negative predictive value, PPV: positive predictive value, FOR: false omission rate, CI: confidence interval.

	Model			
	Female	Male	18-64 years	65+ years
F1-score [%] ↑	30.4 (29.3-31.5)	35.2 (34.0-36.4)	24.3 (23.0-25.7)	36.6 (35.6-37.6)
Sensitivity [%] ↑	44.5 (42.9-46.0)	52.8 (51.1-54.4)	45.6 (43.4-47.9)	49.2 (47.9-50.5)
PPV [%] ↑	23.1 (22.2-24.1)	26.4 (25.3-27.4)	16.6 (15.5-17.5)	29.1 (28.2-30.0)
FOR [%] ↓ (1 - NPV)	0.152 (0.146-0.159)	0.155 (0.147-0.162)	0.058 (0.055-0.062)	0.372 (0.359-0.386)
FPR [%] ↓ (1 - specificity)	0.404 (0.394-0.414)	0.480 (0.467-0.494)	0.246 (0.239-0.254)	0.873 (0.852-0.894)

E.6 ADDITIONAL RESULTS: MODEL WITH PATIENT AGE AND SEX AS EXPLICIT INPUTS

To assess the importance of patient age and sex for accurate model stroke recognition we have performed an experiment to test whether explicitly adding the age and sex of the patient as inputs to the model improves performance.

Specifically, we encode patient sex as two binary numbers which we concatenate to the bag-of-words input representation. The first has the value of 1 if the patient is female, and the second has the value of 1 if the patient is male. Neither is 1 if the patient's sex is unknown/undisclosed. Similarly, we include patient age as a 15-dimensional one-hot vector where the first value represents patients with an age below 25 and the last value represents patients with an age of 90 or above. Values in between represent 5-year intervals.

We trained the model similarly to how we performed our ablation experiments. That is, we used the hyperparameters that gave the best performance on the validation set to train 11 differently seeded versions of the model and then report mean metrics and associated CIs on the MH-1813 test data. The results are as shown in table E.4 below.

We note that the performance difference between the original model and the models with age and sex inputs is statistically insignificant ($p > 0.05$, paired approximate permutation test). We hypothesize this might be because information regarding the age and sex of the patient is, in many cases, already present in the transcript. Another reason might be that the patient's age and sex are less useful

Table E.4: Overall performance on MH-1813 test data for the model that also takes patient age and sex as direct inputs. We also list the original performance of call-takers and the model (w/o sex and age) from the main manuscript for ease of comparison [mean (95% CI)]. NPV: negative predictive value, PPV: positive predictive value, FOR: false omission rate, CI: confidence interval.

	F1-score [%] ↑	Sensitivity [%] ↑	PPV [%] ↑	FOR [%] ↓ (1 - NPV)	FPR [%] ↓ (1 - specificity)
			Overall		
Call-takers	25.8 (23.7-27.9)	52.7 (49.2-56.4)	17.1 (15.5-18.6)	0.105 (0.094-0.116)	0.565 (0.539-0.590)
Model w/o sex and age	35.7 (35.0-36.4)	63.0 (62.0-64.1)	24.9 (24.3-25.5)	0.082 (0.079-0.085)	0.419 (0.413-0.426)
Model w/ sex and age	35.8 (35.1-36.5)	64.1 (63.1-65.1)	24.9 (24.3-25.4)	0.080 (0.077-0.082)	0.427 (0.421-0.434)

discriminators than other indicative factors. This latter hypothesis is supported by our occlusion analysis and table 9.4 where no highly ranked words directly refer to the patient's age or sex.

It is important to note that in practice the information about patient sex and age is extracted from the patient's CPR number which is typically entered by the patient themselves while queuing for MH-1813. However, at call lines in countries without similar systems in place, such information may not be readily available and cannot be incorporated as an explicit input feature. The same is the case for the 1-1-2 call line in Denmark.

E.7 Additional results: Model without MH-1813 training data

The ablation study in the main manuscript that examines the importance of the two different source domains (1-1-2 and MH-1813) includes only two of the three possible combinations of data sources, specifically: training on both 1-1-2 and MH-1813 data, and training only on MH-1813 excluding 1-1-2 data. For this reason, we have conducted an experiment that includes the remaining combination: training only on 1-1-2 excluding MH-1813 data.

Since the manuscript focuses on the MH-1813 line, the main purpose of experimenting with including and excluding the 1-1-2 data was to examine whether using the out-of-domain 1-1-2 data could improve stroke recognition performance of the machine learning framework on the MH-1813 data. This has interest since many prehospital call centres operate both a high acuity emergency line (like 1-1-2) and a low-acuity medical helpline (like MH-1813) which makes high-acuity

data available for modelling and a potentially valuable data source. The ablation experiment performed here, on the other hand, has a different purpose. It aims to show whether a model to assist the MH-1813 helpline could be developed also in the hypothetical case of only having access to out-of-domain, high-acuity training data from 1-1-2. This prospect may be interesting for some call centres that, for instance, have only recently started operating a medical helpline, and so, do not have in-domain training data available.

We trained the models and provided the results of this experiment in table E.5 below using the same methods as in the main manuscript. We tested the significance of these results using the same statistical significance testing methods as used and described in the main manuscript.

We see that the performance of the ensemble model trained only with 1-1-2 data compared to training only with MH-1813 data was worse in terms of sensitivity and FOR ($p < 0.0001$), but on-par in terms of F1-score and PPV, and better in terms of FPR ($p < 0.0001$). Compared to training with all data, training only on 1-1-2 was worse on all metrics ($p < 0.0001$) except FPR, where it was better ($p < 0.0001$). We note that the model still performed better than the call-takers in terms of F1-score, PPV, and FPR ($p < 0.0001$) and was on-par in terms of sensitivity and FOR. All tests for this comparison were paired approximate permutation tests.

In summary, training on only 1-1-2 data was only somewhat worse than training on only MH-1813 data, and still outperformed call-takers to some degree. This indicates that the domain shift between different call lines, even with different acuity levels, is small enough that naive domain transfer of models works well.

Table E.5: Overall performance on MH-1813 test data, performance without 1-1-2 training data, and performance without 1813 training data [mean (95% CI)]. NPV: negative predictive value, PPV: positive predictive value, FOR: false omission rate, CI: confidence interval.

	F1-score [%] ↑	Sensitivity [%] ↑	PPV [%] ↑	FOR [%] ↓ (1 - NPV)	FPR [%] ↓ (1 - specificity)
			Overall		
Call-takers	25.8 (23.7-27.9)	52,7 (49,2-56.4)	17.1 (15.5-18.6)	0.105 (0.094-0.116)	0.565 (0.539-0.590)
Model	35.7 (35.0-36.4)	63.0 (62.0-64.1)	24.9 (24.3-25.5)	0.082 (0.079-0.085)	0.419 (0.413-0.426)
Model w/o 1-1-2	32.4 (31.8-33.1)	60.4 (59.3-61.4)	22.2 (21.6-22.7)	0.088 (0.085-0.091)	0.467 (0.460-0.474)
Model w/o MH-1813	31.4 (30.7-32.1)	50.4 (49.3-51.4)	22.8 (22.2-23.4)	0.110 (0.106-0.113)	0.375 (0.369-0.381)

E.8   Additional results: Detailed model explainability tables

In tables E.6 and E.7 we provide detailed versions of the data presented in the model explainability section in the main manuscript. Specifically, we include the Danish word, its English translation, the rank and impact scores and the number of occurrences of each word.

Table E.6: Mean impact for words with the largest positive rank score in calls predicted as stroke.

	Word, w (*Danish*)	Translation, w (*English*)	Rank, $r^{(w)}$	Count, $D^{(w)}$	Impact, $i^{(d,w)}$ mean ± std.
		Stroke predictions, D = 1,897			
1.	Ambulance	Ambulance	1.000	1,680	0.52 ± 0.51
2.	Blodprop	Blood clot	0.599	895	0.51 ± 0.58
3.	Venstre	Left	0.381	1,108	0.38 ± 0.4
4.	Højre	Right	0.326	1,050	0.31 ± 0.42
5.	Dobbeltsyn	Double vision	0.247	84	1.01 ± 1.26
6.	Ordene	The words	0.217	344	0.6 ± 0.45
7.	Pludselig	Suddenly	0.142	783	0.29 ± 0.28
8.	Arm	Arm	0.140	709	0.3 ± 0.3
9.	Side	Side	0.125	1,139	0.23 ± 0.21
10.	Apopleksi	Stroke	0.102	117	0.33 ± 0.82
11.	Dobbelt	Double	0.102	113	0.54 ± 0.72
12.	Styre	Control	0.092	134	0.63 ± 0.46
13.	Opkald	Call	0.067	39	0.18 ± 1.22
14.	Følelsesløs	Numb	0.065	94	0.53 ± 0.58
15.	Minutter	Minutes	0.064	763	0.22 ± 0.16
16.	Talebesvær	Difficulties speaking	0.063	44	0.87 ± 0.72
17.	Hjerneblødning	Haemorrhagic stroke	0.060	133	0.4 ± 0.49
18.	Hånd	Hand	0.057	297	0.28 ± 0.31
19.	Ambulancen	The ambulance	0.055	521	0.21 ± 0.23
20.	Snøvler	Slurred speech	0.052	58	0.71 ± 0.54
21.	Blodpropper	Blood clots	0.051	224	0.27 ± 0.36
22.	Hurtigt	Fast	0.048	663	0.18 ± 0.18
23.	Udtrykke	Express	0.044	44	0.59 ± 0.74
24.	Blodfortyndende	Blood thinner	0.044	259	0.32 ± 0.22
25.	Usammenhængende	Incoherent	0.043	15	1.14 ± 1.13
26.	Skæv	Lopsided	0.039	211	0.29 ± 0.28
27.	Nedsat	Reduced	0.038	528	0.14 ± 0.21
28.	Hænger	Hangs	0.036	628	0.15 ± 0.17
29.	Forbigående	Transient	0.035	48	0.52 ± 0.62
30.	Vrøvler	Not making sense	0.033	14	1.13 ± 0.89

Table E.7: Mean impact for words with the largest negative rank score in calls predicted as non-stroke.

	Word, w (Danish)	Translation, w (English)	Rank, $r^{(w)}$	Count, $D^{(w)}$	Impact, $i^{(d,w)}$ mean ± std.
			Non-stroke predictions, $D = 342,133$		
1.	Stivkrampe	Tetanus	−1.000	4378	−19.40 ± 10.61
2.	Gravid	Pregnant	−0.901	8749	−12.08 ± 8.64
3.	Skåret	Cut	−0.772	7592	−11.98 ± 8.61
4.	Forbinding	Bandage	−0.569	4561	−12.87 ± 10.08
5.	Amager	Amager (a location)	−0.566	23776	−5.60 ± 4.43
6.	Klokken	O'clock	−0.535	94436	−2.22 ± 2.69
7.	Skadestuen	The emergency room	−0.486	42809	−3.72 ± 3.23
8.	Politiet	The police	−0.413	2903	−10.73 ± 13.77
9.	Hævet	Swollen	−0.388	60559	−2.84 ± 2.38
10.	Håndkøb	over the counter (otc)	−0.372	4641	−11.64 ± 6.00
11.	Halsen	The neck	−0.366	30151	−3.33 ± 3.86
12.	Feber	Fever	−0.361	112586	−1.94 ± 1.76
13.	Recept	Prescription	−0.334	5450	−9.87 ± 5.82
14.	Centimetre	Centimetre	−0.311	12026	−6.01 ± 4.39
15.	Knæet	The knee	−0.300	8875	−6.12 ± 5.91
16.	Apoteket	The pharmacy	−0.267	10085	−6.05 ± 4.49
17.	Maven	The stomach	−0.267	42105	−2.36 ± 2.82
18.	Psykiatrisk	Psychiatric	−0.263	3688	−8.99 ± 8.49
19.	Lungebetændelse	Pneumonia	−0.231	7597	−5.79 ± 5.62
20.	Mavesmerter	Stomach pain	−0.209	10551	−5.12 ± 4.02
21.	Afføring	Stool	−0.199	19155	−3.40 ± 3.27
22.	Ribbenene	The ribs	−0.195	3928	−8.26 ± 6.18
23.	Bløde	Bleed	−0.194	10501	−4.88 ± 3.97
24.	Bløder	Bleeding	−0.193	24313	−2.90 ± 2.93
25.	Ribben	Ribs	−0.189	2941	−8.96 ± 7.56
26.	Brækket	Broken	−0.183	19415	−3.49 ± 2.83
27.	Betændelse	Inflammation	−0.181	10050	−5.27 ± 3.30
28.	Forkølet	Common cold	−0.161	8127	−5.31 ± 3.75
29.	Morgen	Morning or morrow	−0.160	78558	−1.23 ± 1.70
30.	Hævelse	Swelling	−0.159	17762	−3.71 ± 2.32

E.9 Additional results: Fine-tuning of Danish BERT model for stroke recognition

During the preliminary experimental phase, we fine-tuned a BERT model pre-trained on Danish text from CommonCrawl, Wikipedia, OpenSubtitles, and other Danish online forums (available at `https://github.com/certainlyio/nordic_bert`). The model was fine-tuned for 10,000 updates using linear learning rate warm-up (1,000 updates) and decay (9,000 updates). The maximum learning rate was set to 5×10^{-5} and an accumulated batch size of 128. The maximum sequence length of the pre-trained model was 512 input tokens. To accommodate longer input sequences, which was necessary for our dataset, we concatenated several copies of the original positional embedding matrix. The results of the final model are presented in table E.8.

We see that the fine-tuned BERT model performs slightly worse across F1-score, sensitivity, PPV and FOR ($p < 0.0001$, paired approximate permutation test), but better in terms of FPR ($p < 0.0001$, paired approximate permutation test), compared the MLP model presented in the main manuscript. As described in the discussion section, we hypothesize that the number of stroke positives was too small for these advanced models to learn more complex patterns than the MLP ensemble. In addition, the BERT model would likely benefit from pre-training on text data from the target domain, or a domain close to it, rather than various online fora.

Table E.8: Overall performance on MH-1813 test data for the fine-tuned BERT model described in the revised discussion of the paper. Includes performance of the call-takers and the multi-layer perceptron (MLP) from the main manuscript for ease of comparison [mean (95% CI)]. NPV: negative predictive value, PPV: positive predictive value, FOR: false omission rate, CI: confidence interval.

	F1-score [%] ↑	Sensitivity [%] ↑	PPV [%] ↑	FOR [%] ↓ (1 - NPV)	FPR [%] ↓ (1 - specificity)
			Overall		
Call-takers	25.8 (23.7-27.9)	52.7 (49.2-56.4)	17.1 (15.5-18.6)	0.105 (0.094-0.116)	0.565 (0.539-0.590)
MLP	35.7 (35.0-36.4)	63.0 (62.0-64.1)	24.9 (24.3-25.5)	0.082 (0.079-0.085)	0.419 (0.413-0.426)
BERT (fine-tuned)	33.8 (31.5-36.2)	57.5 (53.9-60.9)	23.9 (21.9-25.9)	0.094 (0.084-0.104)	0.403 (0.381-0.424)

E.10 Simulation of a prospective study on 2021 data

E.10.1 Method

The machine learning frameworks can be deployed in different forms in clinical practice. To assess the potential outcomes of deploying this framework in a future prospective study, we performed an experiment using the 2021 test data to simulate different scenarios. Each scenario included two main variables.

I. When is the model prediction presented to the call-taker?

II. How does prediction influence the diagnostic code the call-taker assigns to the call?

There are two primary options as to when the model prediction is presented (I):

1. Notify the call-taker of potential false positive or negative stroke cases after the call ends.

2. Notify the call-taker of potential false positive or negative stroke cases during the call.

Option 1 is identical to the method used in the main study. In option 2, predictions are made during the call based only on partial transcriptions. We implemented option 2 in such a manner that the model predicted every time 50 new words were transcribed and added to the transcript. A stroke positive was triggered only when three consecutive positive predictions were made (i.e., without intermediate negative stroke predictions). In other words, the sigmoid activation of the model had to remain above 0.5 for three consecutive predictions, for example, after 150, 200, and 250 words were transcribed.

As we can only assume how call takers are influenced by model predictions (II), precisely evaluating the hypothetical performance of call takers when supported by a machine learning framework is impossible. Furthermore, option 2 may influence the conversation, further complicating matters. Therefore, we report the results combining the call taker and the model under the following two assumptions:

A. Call-takers change any stroke prediction from negative to positive if the model predicts a positive (call-takers mirror model positives).

B. Call-takers change any stroke prediction from positive to negative if the model predicts a negative (call-takers mirror model negatives).

By definition, method A tended to increase sensitivity and decrease PPV, whereas method B tended to decrease sensitivity and increase PPV.

We also report the results of the model itself (C). This method corresponds to call-takers mirroring the model predictions exactly. This is not feasible in practice, although technically possible, because the conversation and instructions given to patients may conflict with the actions taken by the call-taker after hanging up. Method 1.C is identical to the method employed in the main text, and we have copied the same results here for easier comparison.

E.10.2 Results

As expected (table E.9), method 2.C (raw model predictions during calls) yielded slightly worse results than 1.C (raw model predictions after calls). Compared with method C, method A (call-takers mirror model positives) led to increased sensitivity and decreased PPV, whereas method B (call-takers mirror model negatives) led to decreased sensitivity and increased PPV, as expected. The numerical changes compared with method C are quite large because, in our simulation, the call-taker is assumed to strictly follow methods A or B without divergence.

Method 1.A (call-takers mirror model positives after a call) yields a better F1-score, sensitivity, PPV, and FOR than call-takers alone, although at the cost of a slightly higher FPR. This stands in contrast to methods 1.B, 2.A, and 2.B where either the sensitivity or the PPV is worse for the combined system than for call-takers alone. Regardless, the F1-score (harmonic mean of sensitivity and PPV) is higher for all methods of combining call-takers and model (1.A through 2.C).

These findings highlight that the implementation strategy selected for practice can substantially affect performance. Therefore, it may be possible to implement the system in a way that improves stroke recognition in practice.

E.11 Research in context

E.11.1 Evidence before this study

We searched the PubMed database for articles published in any language up to 9 May 2023 using the following terms: stroke AND (artificial intelligence OR machine learning OR deep learning) AND (EMS OR emergency medical services OR dispatch OR telephone). We identified 88 articles, none of which reported the results of machine learning-based stroke recognition during telephone calls.

One study assessed the potential impact of using speech classification software for stroke recognition by extrapolating results from a similar solution for recognising cardiac arrest. Several authors of this study co-authored the article; however, it did not report the results of an actual novel machine learning framework. The remaining articles primarily reported the use of machine learning in imaging diagnostics, stroke recognition using movement-tracking mobile

Table E.9: Overall performance of model, call-takers and simulated combinations of model and call-takers on MH-1813 test data.

Predictor	Call-taker	Model		Call-taker supported by the model (simulated)			
When	-	After call	During call	After call	During call	After call	During call
Method	-	1.C	2.C	1.A	1.B	2.A	2.B
F1-score [%] ↑	25.8 (23.7-27.9)	35.7 (35.0-36.4)	33.1 (32.4-33.7)	28.9 (28.3-29.5)	33.3 (32.5-34.1)	27.6 (27.0-28.1)	32.7 (31.8-33.5)
Sensitivity [%] ↑	52.7 (49.2-56.4)	63.0 (62.0-64.1)	58.7 (57.7-59.8)	72.4 (71.5-73.3)	43.4 (42.3-44.5)	72.3 (71.4-73.3)	39.1 (38.1-40.1)
PPV [%] ↑	17.1 (15.5-18.6)	24.9 (24.3-25.5)	23.0 (22.5-23.6)	18.0 (17.6-18.4)	27.0 (26.3-27.8)	17.0 (16.7-17.4)	28.1 (27.3-28.9)
FOR [%] ↓ (1 - NPV)	0.105 (0.094-0.116)	0.082 (0.079-0.085)	0.091 (0.088-0.094)	0.061 (0.059-0.064)	0.125 (0.121-0.129)	0.061 (0.059-0.064)	0.134 (0.131-0.138)
FPR [%] ↓ (1 - specificity)	0.565 (0.539-0.590)	0.419 (0.413-0.426)	0.432 (0.426-0.439)	0.726 (0.717-0.735)	0.258 (0.253-0.263)	0.776 (0.767-0.786)	0.221 (0.216-0.226)

devices, and the development of stroke recognition tools using other non-audio data.

E.11.2 Added value of this study

This study is the first to investigate the use of a machine learning framework for stroke recognition in medical helpline calls. The study's results can be replicated in other call lines and for other acute illnesses. A machine learning framework has been previously described for outof-hospital cardiac arrest; however, our results illustrate the feasibility of employing machine learning for detecting stroke - a more complex acute condition.

E.11.3 Implications of all the available evidence

Implementing the framework described in this study could lead to improved recognition of patients with stroke during initial contact with health services. Improving recognition would result in more patients being eligible for advanced stroke treatment and better overall outcomes owing to faster referral to a stroke unit.

E.12 Research in context search term results

The search terms stroke AND (artificial intelligence OR machine learning OR deep learning) AND (EMS OR emergency medical services OR dispatch OR telephone) yielded 88 articles in PubMed. These articles are listed below:

1. Soun J.E., Chow D.S., Nagamine M. et al. Ärtificial Intelligence and Acute Stroke Imaging". *American Journal of Neuroradiology* (2021); 42: 2-11.

2. Murray N.M., Unberath M., Hager G.D., Hui F.K. Ärtificial Intelligence to Diagnose Ischemic Stroke and Identify Large Vessel Occlusions: A Systematic Review". *Journal of Neurointerventional Surgery* (2020); 12: 156-64.

3. Shlobin N.A., Baig A.A., Waqas M. et al. Ärtificial Intelligence for Large-Vessel Occlusion Stroke: A Systematic Review". *World Neurosurgery* (2022); 159: 207-20.e1.

4. Linder S.M., Rosenfeldt A.B., Bay R.C., Sahu K., Wolf S.L., Alberts J.L. Ïmproving Quality of Life and Depression After Stroke Through Telerehabilitation". *The American Journal of Occupational Therapy: Official Publication of the American Occupational Therapy Association* (2015); 69: 6902290020p1-10.

5. Humphreys K., Shover C.L., Andrews C.M. et al. "Responding to the Opioid Crisis in North America and Beyond: Recommendations of the Stanford-Lancet Commission". *The Lancet (London, England)* (2022); 399: 555-604.

6. Fassbender K., Lesmeister M., Merzou F. "Prehospital Stroke Management and Mobile Stroke Units". *Current Opinion in Neurology* (2023); 36: 140-6.

7. McDonough R.V., Ospel J.M., Majoie C.B.L.M. et al. "Clinical Outcome of Patients with Mild Pre-Stroke Morbidity Following Endovascular Treatment: A HERMES Substudy". *Journal of Neurointerventional Surgery* (2023); 15: 214-20.

8. Bat-Erdene B.O., Saver J.L. Äutomatic Acute Stroke Symptom Detection and Emergency Medical Systems Alerting by Mobile Health Technologies: A Review". *Journal of Stroke and Cerebrovascular Diseases: The Official Journal of National Stroke Association* (2021); 30: 105826.

9. Bonini N., Vitolo M., Imberti J.F. et al. "Mobile Health Technology in Atrial Fibrillation". *Expert Review of Medical Devices* (2022); 19: 327-40.

10. Scholz M.L., Collatz-Christensen H., Blomberg S.N.F., Boebel S., Verhoeven J., Krafft T. Ärtificial Intelligence in Emergency Medical Services Dispatching: Assessing the Potential Impact of an Automatic Speech Recognition Software on Stroke Detection Taking the Capital Region of Denmark as Case in Point". *Scandinavian Journal of Trauma, Resuscitation, and Emergency Medicine* (2022); 30: 36.

11. Gupta R., Krishnam S.P., Schaefer P.W., Lev M.H., Gonzalez R.G. Än East Coast Perspective on Artificial Intelligence and Machine Learning: Part 2 - Ischemic Stroke Imaging and Triage". *Neuroimaging Clinics of North America* (2020); 30: 467-78.

12. Esbjörnsson M., Ullberg T. SSafety and Usability of Wearable Accelerometers for Stroke Detection: The STROKE ALARM PRO 1 Study". *Journal of Stroke and Cerebrovascular Diseases: The Official Journal of National Stroke Association* (2022); 31: 106762.

13. Fast L., Temuulen U., Villringer K. et al. "Machine Learning-Based Prediction of Clinical Outcomes After First-Ever Ischemic Stroke". *Frontiers in Neurology* (2023); 14: 1114360.

14. Gupta R., Krishnam S.P., Schaefer P.W., Lev M.H., Gilberto Gonzalez R. Än East Coast Perspective on Artificial Intelligence and Machine Learning: Part 1 - Hemorrhagic Stroke Imaging and Triage". *Neuroimaging Clinics of North America* (2020); 30: 459-66.

15. Cai T., Ni H., Yu M. et al. "DeepStroke: An Efficient Stroke Screening Framework for Emergency Rooms with Multimodal Adversarial Deep Learning". *Medical Image Analysis* (2022); 80: 102522.

16. Zachrison K.S., Dhand A., Schwamm L.H., Onnela J.P. Ä Network Approach to Stroke Systems of Care". *Circulation. Cardiovascular Quality and Outcomes* (2019); 12: e005526.

17. O'Connell G.C., Walsh K.B., Smothers C.G. et al. Üse of Deep Artificial Neural Networks to Identify Stroke During Triage via Subtle Changes in Circulating Cell Counts". *BMC Neurology* (2022); 22: 206.

18. Yang L., Liu Q., Zhao Q., Zhu X., Wang L. "Machine Learning Is a Valid Method for Predicting Prehospital Delay After Acute Ischemic Stroke". *Brain and Behavior* (2020); 10: e01794.

19. Aldridge C.M., McDonald M.M., Wruble M. et al. "Human vs. Machine Learning-Based Detection of Facial Weakness Using Video Analysis". *Frontiers in Neurology* (2022); 13: 878282.

20. Lachance C.C., Ford C. "Portable Stroke Detection Devices for Patients with Stroke Symptoms: A Review of Diagnostic Accuracy and Cost-Effectiveness". *Canadian Agency for Drugs and Technologies in Health* (2019).

21. Fardoun H.M., Mashat A.S. "Methodologies, Models, and Algorithms for Patients Rehabilitation". *Methods of Information in Medicine* (2016); 55: 60-4.

22. Wibring K., Magnusson C., Axelsson C., Lundgren P., Herlitz J., Andersson Hagiwara M. "Towards Definitions of Time-Sensitive Conditions in Prehospital Care". *Scandinavian Journal of Trauma, Resuscitation, and Emergency Medicine* (2020); 28: 7.

23. Mayampurath A., Parnianpour Z., Richards C.T. et al. "Improving Prehospital Stroke Diagnosis Using Natural Language Processing of Paramedic Reports". *Stroke* (2021); 52: 2676-9.

24. Morey J.R., Zhang X., Yaeger K.A. et al. "Real-World Experience with Artificial Intelligence-Based Triage in Transferred Large Vessel Occlusion Stroke Patients". *Cerebrovascular Diseases (Basel, Switzerland)* (2021); 50: 450-5.

25. Sung S.F., Hung L.C., Hu Y.H. "Developing a Stroke Alert Trigger for Clinical Decision Support at Emergency Triage Using Machine Learning". *International Journal of Medical Informatics* (2021); 152: 104505.

26. Zhang Z., Zhou D., Zhang J. et al. "Multilayer Perceptron-Based Prediction of Stroke Mimics in Prehospital Triage". *Scientific Reports* (2022); 12: 17994.

27. Park E., Lee K., Han T., Nam H.S. "Automatic Grading of Stroke Symptoms for Rapid Assessment Using Optimized Machine Learning and 4-Limb Kinematics: Clinical Validation Study". *Journal of Medical Internet Research* (2020); 22: e20641.

28. Chen M., Tan X., Padman R. "A Machine Learning Approach to Support Urgent Stroke Triage Using Administrative Data and Social Determinants of Health at Hospital Presentation: Retrospective Study". *Journal of Medical Internet Research* (2023); 25: e36477.

29. Uchida K., Kouno J., Yoshimura S. et al. "Development of Machine Learning Models to Predict Probabilities and Types of Stroke at Prehospital Stage: The Japan Urgent Stroke Triage Score Using Machine Learning (JUST-ML)". *Translational Stroke Research* (2022); 13: 370-81.

30. Finck T., Schinz D., Grundl L. et al. "Automated Detection of Ischemic Stroke and Subsequent Patient Triage in Routinely Acquired Head CT". *Clinical Neuroradiology* (2022); 32: 419-26.

31. Sangal R.B., Fodeh S., Taylor A. et al. "Identification of Patients with Nontraumatic Intracranial Hemorrhage Using Administrative Claims Data". *Journal of Stroke and Cerebrovascular Diseases: The Official Journal of National Stroke Association* (2020); 29: 105306.

32. Adedinsewo D., Carter R.E., Attia Z. et al. "Artificial Intelligence-Enabled ECG Algorithm to Identify Patients with Left Ventricular Systolic Dysfunction Presenting to the Emergency Department with Dyspnea". *Circulation. Arrhythmia and Electrophysiology* (2020); 13: e008437.

33. Liu L., Yu B., Han M., Yuan S., Wang N. "Mild Cognitive Impairment Understanding: An Empirical Study by Data-Driven Approach". *BMC Bioinformatics* (2019); 20: 481.

34. Hayashi Y., Shimada T., Hattori N. et al. "A Prehospital Diagnostic Algorithm for Strokes Using Machine Learning: A Prospective Observational Study". *Scientific Reports* (2021); 11: 20519.

35. Grant L., Joo P., Nemnom M.J., Thiruganasambandamoorthy V. "Machine Learning Versus Traditional Methods for the Development of Risk Stratification Scores: A Case Study Using Original Canadian Syncope Risk Score Data". *Internal and Emergency Medicine* (2022); 17: 1145-53.

36. van Meenen L.C.C., van Stigt M.N., Marquering H.A. et al. "Detection of Large Vessel Occlusion Stroke with Electroencephalography in the Emergency Room: First Results of the ELECTRA-STROKE Study". *Journal of Neurology* (2022); 269: 2030-8.

37. van Meenen L.C.C., den Hartog S.J., Groot A.E. et al. "Relationship Between Primary Stroke Center Volume and Time to Endovascular Thrombectomy in Acute Ischemic Stroke". *European Journal of Neurology* (2021); 28: 4031-8.

38. Chien H.C., Yang T.L., Juang W.C., Chen Y.A., Li Y.J., Chen C.Y. "Pilot Report for Intracranial Hemorrhage Detection with Deep Learning Implanted Head Computed Tomography Images at Emergency Department". *Journal of Medical Systems* (2022); 46: 49.

39. Huo D., Leppert M., Pollard R. et al. "Large Vessel Occlusion Prediction in the Emergency Department with National Institutes of Health Stroke Scale Components: A Machine Learning Approach". *Journal of Stroke and Cerebrovascular Diseases: The Official Journal of National Stroke Association* (2021); 30: 106030.

40. Wu M.R., Chen Y.T., Li Z.X. et al. "Dysphagia Screening and Pneumonia After Subarachnoid Hemorrhage: Findings from the Chinese Stroke Center Alliance". *CNS Neuroscience & Therapeutics* (2022); 28: 913-21.

41. Bacchi S., Oakden-Rayner L., Zerner T., Kleinig T., Patel S., Jannes J. "Deep Learning Natural Language Processing Successfully Predicts the Cerebrovascular Cause of Transient Ischemic Attack-Like Presentations". *Stroke* (2019); 50: 758-60.

42. Heldt F.S., Vizcaychipi M.P., Peacock S. et al. "Early Risk Assessment for COVID-19 Patients from Emergency Department Data Using Machine Learning". *Scientific Reports* (2021); 11: 4200.

43. Lu Z., Xiong Y., Yang K. et al. "What Predicts Large Vessel Occlusion in Mild Stroke Patients". *BMC Neurology* (2023); 23: 29.

44. Marijon E., Garcia R., Narayanan K., Karam N., Jouven X. "Fighting Against Sudden Cardiac Death: Need for a Paradigm Shift-Adding Near-Term Prevention and Pre-Emptive Action to Long-Term Prevention". *European Heart Journal* (2022); 43: 1457-64.

45. You J., Yu P.L.H., Tsang A.C.O. et al. "3D Dissimilar-Siamese-U-Net for Hyperdense Middle Cerebral Artery Sign Segmentation". *Computerized Medical Imaging*

and Graphics: The Official Journal of the Computerized Medical Imaging Society (2021); 90: 101898.

46. Moral-Munoz J.A., Zhang W., Cobo M.J., Herrera-Viedma E., Kaber D.B. SSmartphone-Based Systems for Physical Rehabilitation Applications: A Systematic Review". *Assistive Technology: The Official Journal of RESNA* (2021); 33: 223-36.

47. Salman O.H., Aal-Nouman M.I., Taha Z.K., Alsabah M.Q., Hussein Y.S., Abdelkareem Z.A. "Formulating Multi-Diseases Dataset for Identifying, Triaging, and Prioritizing Patients to Multi-Medical Emergency Levels: Simulated Dataset Accompanied with Codes". *Data in Brief* (2021); 34: 106576.

48. Desai A., Zumbo A., Giordano M. et al. "Word2Vec Word Embedding-Based Artificial Intelligence Model in the Triage of Patients with Suspected Diagnosis of Major Ischemic Stroke: A Feasibility Study". *International Journal of Environmental Research and Public Health* (2022); 19: 15295.

49. Rinkel L.A., Prick J.C.M., Slot R.E.R. et al. Ïmpact of the COVID-19 Outbreak on Acute Stroke Care". *Journal of Neurology* (2021); 268: 403-8.

50. Chilamkurthy S., Ghosh R., Tanamala S. et al. "Deep Learning Algorithms for Detection of Critical Findings in Head CT Scans: A Retrospective Study". *The Lancet (London, England)* (2018); 392: 2388-96.

51. Kramer N.M., Demaerschalk B.M. Ä Novel Application of Teleneurology: Robotic Telepresence in Supervision of Neurology Trainees". *Telemedicine Journal and E-Health: The Official Journal of the American Telemedicine Association* (2014); 20: 1087-92.

52. Yadav K., Sarioglu E., Choi H.A., Cartwright W.B., Hinds P.S., Chamberlain J.M. Ä̈utomated Outcome Classification of Computed Tomography Imaging Reports for Pediatric Traumatic Brain Injury". *Academic Emergency Medicine: Official Journal of the Society for Academic Emergency Medicine* (2016); 23: 171-8.

53. van Meenen L.C.C., Riedijk F., Stolp J. et al. "Pre- and Interhospital Workflow Times for Patients with Large Vessel Occlusion Stroke Transferred for Endovascular Thrombectomy". *Frontiers in Neurology* (2021); 12: 730250.

54. Saberi-Movahed F., Mohammadifard M., Mehrpooya A. et al. "Decoding Clinical Biomarker Space of COVID-19: Exploring Matrix Factorization-Based Feature Selection Methods". *Computers in Biology and Medicine* (2022); 146: 105426.

55. Shimada T., Matsubara K., Koyama D. et al. "Development of Evaluation System for Cerebral Artery Occlusion in Emergency Medical Services: Noninvasive Measurement and Utilization of Pulse Waves". *Scientific Reports* (2023); 13: 3339.

56. Gil-Jardiné C., Chenais G., Pradeau C. et al. "Trends in Reasons for Emergency Calls During the COVID-19 Crisis in the Department of Gironde, France Using Artificial Neural Network for Natural Language Classification". *Scandinavian Journal of Trauma, Resuscitation and Emergency Medicine* (2021); 29: 55.

57. Santala O.E., Lipponen J.A., Jäntti H. et al. "Continuous mHealth Patch Monitoring for the Algorithm-Based Detection of Atrial Fibrillation: Feasibility and Diagnostic Accuracy Study". *JMIR Cardio* (2022); 6: e31230.

58. O'Brien M.K., Shawen N., Mummidisetty C.K. et al. "Activity Recognition for Persons with Stroke Using Mobile Phone Technology: Toward Improved Performance in a Home Setting". *Journal of Medical Internet Research* (2017); 19: e184.

59. Bateman R.M., Sharpe M.D., Jagger J.E. et al. "36th International Symposium on Intensive Care and Emergency Medicine: Brussels, Belgium. 15-18 March 2016". *Critical Care (London, England)* (2016); 20: 94.

60. Modica C.M., Johnson B.R., Zalewski C. et al. "Hearing Loss and Irritability Reporting Without Vestibular Differences in Explosive Breaching Professionals". *Frontiers in Neurology* (2020); 11: 588377.

61. Palmcrantz S., Plantin J., Borg J. "Factors Affecting the Usability of an Assistive Soft Robotic Glove after Stroke or Multiple Sclerosis". *Journal of Rehabilitation Medicine* (2020); 52: jrm00027.

62. Mitsopoulos K., Fiska V., Tagaras K. et al. "NeuroSuitUp: System Architecture and Validation of a Motor Rehabilitation Wearable Robotics and Serious Game Platform". *Sensors (Basel, Switzerland)* (2023); 23: 3281.

63. Meng G., Tan Y., Fang M., Yang H., Liu X., Zhao Y. "Meteorological Factors Related to Emergency Admission of Elderly Stroke Patients in Shanghai: Analysis with a Multilayer Perceptron Neural Network". *Medical Science Monitor: International Medical Journal of Experimental and Clinical Research* (2015); 21: 3600-7.

64. Iglesias J.E., Schleicher R., Laguna S. et al. "Quantitative Brain Morphometry of Portable Low-Field-Strength MRI Using Super-Resolution Machine Learning". *Radiology* (2023); 306: e220522.

65. Lai F. "Stroke Networks Based on Robotic Telepresence". *Journal of Telemedicine and Telecare* (2009); 15: 135-6.

66. Kim D., Oh J., Im H., Yoon M., Park J., Lee J. "Automatic Classification of the Korean Triage Acuity Scale in Simulated Emergency Rooms Using Speech Recognition and Natural Language Processing: A Proof of Concept Study". *Journal of Korean Medical Science* (2021); 36: e175.

67. Jonsson A., Binongo J., Patel P. et al. "Mastering the Learning Curve for Robotic-Assisted Coronary Artery Bypass Surgery". *The Annals of Thoracic Surgery* (2023); 115: 1118-25.

68. Knowlton S.M., Sencan I., Aytar Y. et al. "Sickle Cell Detection Using a Smartphone". *Scientific Reports* (2015); 5: 15022.

69. Che F., Liu Y., Gong X. et al. "Extracranial Carotid Plaque Hemorrhage Is Independently Associated with Poor 3-Month Functional Outcome after Acute Ischemic Stroke-A Prospective Cohort Study". *Frontiers in Neurology* (2021); 12: 780436.

70. Boers A.M.M., Jansen I.G.H., Brown S. et al. "Mediation of the Relationship Between Endovascular Therapy and Functional Outcome by Follow-Up Infarct Volume in Patients with Acute Ischemic Stroke". *JAMA Neurology* (2019); 76: 194-202.

71. Stone J.R., Avants B.B., Tustison N.J. et al. "Functional and Structural Neuroimaging Correlates of Repetitive Low-Level Blast Exposure in Career Breachers". *Journal of Neurotrauma* (2020); 37: 2468-81.

72. Rickards C.A., Vyas N., Ryan K.L. et al. Äre You Bleeding? Validation of a Machine-Learning Algorithm for Determination of Blood Volume Status: Application to Remote Triage". *Journal of Applied Physiology (Bethesda, Md. : 1985)* (2014); 116: 486-94.

73. Edwards K.A., Greer K., Leete J. et al. "Neuronally-Derived Tau Is Increased in Experienced Breachers and Is Associated with Neurobehavioral Symptoms". *Scientific Reports* (2021); 11: 19527.

74. Cheng A.L., Liu J., Bravo S., Miller J.C., Pahlevan N.M. SScreening Left Ventricular Systolic Dysfunction in Children Using Intrinsic Frequencies of Carotid Pressure Waveforms Measured by a Novel Smartphone-Based Device". *Physiological Measurement* (2023); 44.

75. Grunwald I.Q., Kulikovski J., Reith W. et al. "Collateral Automation for Triage in Stroke: Evaluating Automated Scoring of Collaterals in Acute Stroke on Computed Tomography Scans". *Cerebrovascular Diseases (Basel, Switzerland)* (2019); 47: 217-22.

76. Hyakutake K., Morishita T., Saita K. et al. Ëffects of Home-Based Robotic Therapy Involving the Single-Joint Hybrid Assistive Limb Robotic Suit in the Chronic Phase of Stroke: A Pilot Study". *BioMed Research International* (2019); 2019: 5462694.

77. Murray C., Ortiz E., Kubin C. Äpplication of a Robot for Critical Care Rounding in Small Rural Hospitals". *Critical Care Nursing Clinics of North America* (2014); 26: 477-85.

78. Edwards K.A., Leete J.J., Tschiffely A.E. et al. "Blast Exposure Results in Tau and Neurofilament Light Chain Changes in Peripheral Blood". *Brain Injury* (2020); 34: 1213-21.

79. Oresko J.J., Duschl H., Cheng A.C. Ä Wearable Smartphone-Based Platform for Real-Time Cardiovascular Disease Detection via Electrocardiogram Processing". *IEEE Transactions on Information Technology in Biomedicine: A Publication of the IEEE Engineering in Medicine and Biology Society* (2010); 14: 734-40.

80. Alcoceba-Herrero I., Coco-Martín M.B., Leal-Vega L. et al. "Randomized Controlled Trial Evaluating the Benefit of a Novel Clinical Decision Support System for the Management of COVID-19 Patients in Home Quarantine: A Study Protocol". *International Journal of Environmental Research and Public Health* (2023); 20: 2300.

81. van Meenen L.C.C., Groot A.E., Venema E. et al. Ïnterhospital Transfer vs. Direct Presentation of Patients with a Large Vessel Occlusion Not Eligible for IV Thrombolysis". *Journal of Neurology* (2020); 267: 2142-50.

82. Forghani R., Gupta R. "Case of the Season: Artificial Intelligence in Clinical Practice-Large Vessel Occlusion Triage in Stroke Imaging". *Seminars in Roentgenology* (2023); 58: 147-51.

83. Jabbour P., Gonzalez L.F., Tjoumakaris S., Randazzo C., Rosenwasser R. SStroke in the Robotic Era". *World Neurosurgery* (2010); 73: 603-4.

84. Larkin H. Ön the Spot When Doc Is Not". *Hospitals & Health Networks* (2008); 82: 41-2.

85. Jiang Y., Guarino P., Ma S. et al. "Bayesian Accrual Prediction for Interim Review of Clinical Studies: Open Source R Package and Smartphone Application". *Trials* (2016); 17: 336.

86. Lee J.H., Oh B.J., Ahn J.Y. et al. Ëffectiveness of Automatic Acute Stroke Alert System Based on UMLS Mapped Local Terminology Codes at Emergency Department". *AMIA ... Annual Symposium Proceedings. AMIA Symposium* (2008); 1018.

87. "Robot Saves ED Stroke Patients, Addresses Subspecialist Shortage". *ED Management: The Monthly Update on Emergency Department Management* (2011); 23: 13-5.

88. Nagy Z., Simon P., Sipos E., Kozmann G. "The Main Elements of the Information System of the National Stroke Program (Smart Card - Telecommunication - Knowledge Bases)". *Medinfo. MEDINFO* (1995); 8 Pt 2: 1496-9.

List of Figures

List of Tables

BIBLIOGRAPHY

[1] Ahmad, M. A., Eckert, C., Teredesai, A., "Interpretable Machine Learning in Healthcare". In: *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. 2018 (cited on page 3).

[2] Ahmadian, A., Lindsten, F., "Likelihood-Free out-of-Distribution Detection with Invertible Generative Models". In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. International Joint Conferences on Artificial Intelligence Organization, 2021 (cited on pages 24, 30, 60, 69, 272, 273).

[3] Aksan, E., Hilliges, O., "STCN: Stochastic Temporal Convolutional Networks". In: *Proceedings of the 7th International Conference on Learning Representations (ICLR)*. New Orleans, LA, USA, 2019 (cited on pages 85, 92, 93, 98, 99, 102, 104, 106, 185, 242, 243, 246, 249, 251, 254).

[4] Aleksic, P. S., Katsaggelos, A. K., "Audio-Visual Biometrics". In: *Proceedings of the IEEE* 94.11 (2006) (cited on page 197).

[5] Alemi, A. A., Fischer, I., Dillon, J. V., *Uncertainty in the Variational Information Bottleneck*. 2018. arXiv: 1807.00906 (cited on pages 47, 51, 53).

[6] Alemi, A. A., Fischer, I., Dillon, J. V., Murphy, K., "Deep Variational Information Bottleneck". In: *Proceedings of the 5th International Conference on Learning Representations (ICLR)*. Toulon, France, 2017 (cited on page 26).

[7] Alemi, A. A., Poole, B., Fischer, I., Dillon, J. V., Saurous, R. A., Murphy, K., "Fixing a Broken ELBO". In: *Proceedings of the International Conference on Machine Learning (ICML)*. Stockholm, Sweden: PMLR, 2018. arXiv: 1711.00464 (cited on pages 32, 158).

[8] Andrew, G., Arora, R., Bilmes, J. A., Livescu, K., "Deep Canonical Correlation Analysis". In: *Proceedings of the 30th International Conference on Machine Learning*. 2013 (cited on page 198).

[9] Anguera, X., Rodriguez-Fuentes, L.-J., Buzo, A., Metze, F., Szöke, I., Penagarikano, M., "QUESST2014: Evaluating Query-by-Example Speech Search in a Zero-Resource Setting with Real-Life Queries". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015 (cited on page 203).

[10] Ardila, R., Branson, M., Davis, K., Kohler, M., Meyer, J., Henretty, M., Morais, R., Saunders, L., Tyers, F., Weber, G., "Common Voice: A Massively-Multilingual Speech Corpus". In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*. 2020 (cited on page 202).

[11] Arjovsky, M., Chintala, S., Bottou, L., *Wasserstein GAN*. 2017. arXiv: 1701.07875. (Visited on 12 April 2023) (cited on pages 30, 217).

[12] Arpit, D., Jastrzębski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., Lacoste-Julien, S., "A Closer Look at Memorization in Deep Networks". In: (2017) (cited on page 11).

[13] Artetxe, M., Labaka, G., Agirre, E., "A Robust Self-Learning Method for Fully Unsupervised Cross-Lingual Mappings of Word Embeddings". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*. Melbourne, Australia: Association for Computational Linguistics, 2018 (cited on page 218).

[14] Artetxe, M., Labaka, G., Agirre, E., Cho, K., "Unsupervised Neural Machine Translation". In: *International Conference on Learning Representations (ICLR)* (2018) (cited on page 219).

[15]   Astudillo, R. F., Kolossa, D., Mandelartz, P., Orglmeister, R., "An Uncertainty Propagation Approach to Robust ASR Using the ETSI Advanced Front-End". In: *IEEE Journal of Selected Topics in Signal Processing* 4.5 (2010) (cited on page 11).

[16]   Astudillo, R. F., Kolossa, D., Abad, A., Zeiler, S., Saeidi, R., Mowlaee, P., Silva Neto, J. P., Martin, R., "Integration of Beamforming and Uncertainty-of-Observation Techniques for Robust ASR in Multi-Source Environments". In: *Computer Speech & Language* 27.3 (2013). ISSN: 0885-2308 (cited on page 11).

[17]   Babu, A., Wang, C., Tjandra, A., Lakhotia, K., Xu, Q., Goyal, N., Singh, K., Platen, P., Saraf, Y., Pino, J., Baevski, A., Conneau, A., Auli, M., *XLS-R: Self-supervised Cross-Lingual Speech Representation Learning at Scale*. 2021. arXiv: 2111.09296 (cited on pages 202, 205, 207, 208).

[18]   Bach, F. R., Jordan, M. I., *A Probabilistic Interpretation of Canonical Correlation Analysis*. 688. Department of Statistics, University of California, Berkeley, 2005 (cited on page 198).

[19]   Badino, L., Canevari, C., Fadiga, L., Metta, G., "Integrating Articulatory Data in Deep Neural Network-Based Acoustic Modeling". In: *Comp. Sp. & Lang.* 36 (2016) (cited on page 198).

[20]   Badino, L., Canevari, C., Fadiga, L., Metta, G., "An Auto-Encoder Based Approach to Unsupervised Learning of Subword Units". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014 (cited on pages 82, 186).

[21]   Badino, L., Mereta, A., Rosasco, L., "Discovering Discrete Subword Units with Binarized Autoencoders and Hidden-Markov-model Encoders". In: *Sixteenth Annual Conference of the International Speech Communication Association*. 2015 (cited on page 186).

[22]   Baevski, A., Auli, M., Mohamed, A., "Effectiveness of Self-Supervised Pre-Training for Speech Recognition". 2020. arXiv: 1911.03912 (cited on pages 87, 190, 194, 207).

[23]   Baevski, A., Hsu, W.-N., CONNEAU, A., Auli, M., "Unsupervised Speech Recognition". In: *Advances in Neural Information Processing Systems*. Volume 34. Curran Associates, Inc., 2021 (cited on pages 81, 213, 217–220).

[24]   Baevski, A., Hsu, W.-N., Xu, Q., Babu, A., Gu, J., Auli, M., *Data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language*. Facebook AI Research blog, 2022 (cited on pages 157, 190, 196).

[25]   Baevski, A., Schneider, S., Auli, M., "Vq-Wav2vec: Self-Supervised Learning of Discrete Speech Representations". 2020. arXiv: 1910.05453 (cited on pages 87, 190, 194, 206).

[26]   Baevski, A., Zhou, H., Mohamed, A., Auli, M., "Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations". In: *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*. Virtual, 2020. arXiv: 2006.11477 (cited on pages 79, 83, 85, 92, 95, 137, 157, 161, 162, 171, 190, 207, 208, 218, 219).

[27]   Bai, T., Vucetic, S., "Improving Medical Code Prediction from Clinical Text via Incorporating Online Knowledge Sources". In: *The World Wide Web Conference*. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019. ISBN: 978-1-4503-6674-8 (cited on page 121).

[28]   Baluenfeldt, R., Wienecke, T., Dansk Neurologisk Selskab, *National Neurologisk Behandlingsvejledning: Iskæmisk Apopleksi - Akut Udredning Og Behandling*. 2021 (cited on page 148).

[29]   Bansal, S., Kamper, H., Lopez, A., Goldwater, S., *Towards Speech-to-Text Translation without Speech Recognition*. 2017. arXiv: 1702.03856. (Visited on 23 September 2023) (cited on page 95).

[30]   Bao, W., Lin, H., Zhang, Y., Wang, J., Zhang, S., "Medical Code Prediction via Capsule Networks and ICD Knowledge". In: *BMC Medical Informatics and Decision Making* 21.2 (2021). ISSN: 1472-6947 (cited on pages 116, 120, 121, 171).

[31] Bapna, A., Chung, Y.-a., Wu, N., Gulati, A., Jia, Y., Clark, J. H., Johnson, M., Riesa, J., Conneau, A., Zhang, Y., *SLAM: A Unified Encoder for Speech and Language Modeling via Speech-Text Joint Pre-Training*. 2021. arXiv: 2110.10329 (cited on page 199).

[32] Barker, J., Watanabe, S., Vincent, E., Trmal, J., "The Fifth 'CHiME' Speech Separation and Recognition Challenge: Dataset, Task and Baselines". In: *Annual Conference of the International Speech Communication Association*. 2018 (cited on page 203).

[33] Bartholomew, D. J., Knott, M., Moustaki, I., *Latent Variable Models and Factor Analysis: A Unified Approach*. 3rd ed. Wiley Series in Probability and Statistics. Chichester, West Sussex: Wiley, 2011. 277 pages. ISBN: 978-0-470-97192-5 978-1-119-97059-0 978-1-119-97058-3 978-1-119-97370-6 978-1-119-97371-3 (cited on pages 89, 92).

[34] Baskar, M. K., Watanabe, S., Astudillo, R. F., Hori, T., Burget, L., Cernocký, J., "Semi-Supervised Sequence-to-Sequence ASR Using Unpaired Speech and Text". In: *Proceedings of the Conference of the International Speech Communication Association (Interspeech)*. ISCA, 2019 (cited on page 221).

[35] Bauer, M., Mnih, A., "Generalized Doubly-Reparameterized Gradient Estimators". In: *3rd Symposium on Advances in Approximate Bayesian Inference*. 2021. arXiv: 2101.11046 (cited on pages 36, 161, 170).

[36] Bayer, J., Soelch, M., Mirchev, A., Kayalibay, B., Smagt, P., "Mind the Gap When Conditioning Amortised Inference in Sequential Latent-Variable Models". In: *Proceedings of the 9th International Conference on Learning Representations (ICLR)*. Virtual, 2021 (cited on page 102).

[37] Beijing DataTang Technology Co., Ltd. *Aidatatang 200zh, a Free Chinese Mandarin Speech Corpus* (cited on page 202).

[38] Bell, P., Fainberg, J., Klejch, O., Li, J., Renals, S., Swietojanski, P., "Adaptation Algorithms for Neural Network-Based Speech Recognition: An Overview". In: *IEEE Open Journal of Signal Processing* 2 (2021) (cited on pages 179, 181).

[39] Bengio, E., Bacon, P.-L., Pineau, J., Precup, D., "Conditional Computation in Neural Networks for Faster Models". 2016. arXiv: 1511.06297 (cited on page 225).

[40] Bengio, S., Heigold, G., "Word Embeddings for Speech Recognition". In: *Proceedings of the 15th Conference of the International Speech Communication Association (Interspeech)*. 2014 (cited on page 200).

[41] Bengio, Y., Courville, A. C., Vincent, P., "Representation Learning: A Review and New Perspectives". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013) (cited on pages 43, 79, 177).

[42] Bengio, Y., Léonard, N., Courville, A., *Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation*. 2013. arXiv: 1308.3432 (cited on pages 86, 185).

[43] Benjamini, Y., Hochberg, Y., "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing". In: *Journal of the Royal statistical society: series B (Methodological)* 57.1 (1995) (cited on pages 72, 73).

[44] Bennett, P. N. "Using Asymmetric Distributions to Improve Text Classifier Probability Estimates". In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*. 2003 (cited on page 11).

[45] Bergamin, F., Mattei, P.-A., **Havtorn, J. D.**, Senetaire, H., Schmutz, H., Maaløe, L., Hauberg, S., Frellsen, J., "Model-Agnostic Out-of-Distribution Detection Using Combined Statistical Tests". In: *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*. Volume 151. Valencia, Span: PMLR, 2022. arXiv: 2203.01097 (cited on pages vii, 13, 24, 29, 59).

[46] Berge, E., Whiteley, W., Audebert, H., De Marchis, G. M., Fonseca, A. C., Padiglioni, C., Pérez de la Ossa, N., Strbian, D., Tsivgoulis, G., Turc, G., "European Stroke Organisation (ESO) Guidelines on Intravenous Thrombolysis for Acute Ischaemic Stroke". In: *European Stroke Journal* 6.1 (2021) (cited on pages 8, 139).

[47] Bergman, L., Hoshen, Y., "Classification-Based Anomaly Detection for General Data". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. Addis Ababa, Ethiopia, 2020 (cited on pages 24, 30, 157).

[48] Bhati, S., Villalba, J., Żelasko, P., Moro-Velazquez, L., Dehak, N., *Segmental Contrastive Predictive Coding for Unsupervised Word Segmentation*. 2021. arXiv: 2106.02170. (Visited on 23 September 2023) (cited on pages 85, 89).

[49] Bhati, S., Villalba, J., Żelasko, P., Moro-Velazquez, L., Dehak, N., "Unsupervised Speech Segmentation and Variable Rate Representation Learning Using Segmental Contrastive Predictive Coding". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 30 (2022) (cited on page 89).

[50] Bishop, C. M. "Novelty Detection and Neural-Network Validation". In: *IEE Proceedings - Vision, Image and Signal Processing* 141.4 (1994). ISSN: 1350245x, 13597108 (cited on pages 22, 24, 28, 42, 43, 59).

[51] Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006. ISBN: 978-0-387-31073-2 (cited on page 105).

[52] Blomberg, S. N., Christensen, H. C., Lippert, F., Ersbøll, A. K., Torp-Petersen, C., Sayre, M. R., Kudenchuk, P. J., Folke, F., "Effect of Machine Learning on Dispatcher Recognition of Out-of-Hospital Cardiac Arrest during Calls to Emergency Medical Services: A Randomized Clinical Trial". In: *JAMA Network Open* 4.1 (2021) (cited on pages 5, 8, 9, 140, 144, 166, 172).

[53] Blomberg, S. N., Folke, F., Ersbøll, A. K., Christensen, H. C., Torp-Pedersen, C., Sayre, M. R., Counts, C. R., Lippert, F. K., "Machine Learning as a Supportive Tool to Recognize Cardiac Arrest in Emergency Calls". In: *Resuscitation* 138 (2019) (cited on pages 5, 8, 9, 140, 147, 151, 166, 172).

[54] Bohm, K., Kurland, L., "The Accuracy of Medical Dispatch - A Systematic Review". In: *Scandinavian Journal of Trauma, Resuscitation and Emergency Medicine* 26 (2018) (cited on pages 8, 140).

[55] Bommasani, R. *On the Opportunities and Risks of Foundation Models*. 2021. arXiv: 2108.07258 (cited on page 178).

[56] Borgholt, L., **Havtorn, J. D.**, Abdou, M., Edin, J., Maaløe, L., Søgaard, A., Igel, C., *Do We Still Need Automatic Speech Recognition for Spoken Language Understanding?* 2021. arXiv: 2111.14842 (cited on pages viii, 95).

[57] Borgholt, L., **Havtorn, J. D.**, Agić, Ž., Søgaard, A., Maaløe, L., Igel, C., "Do End-to-End Speech Recognition Models Care about Context?" In: *Proceedings of the 21st Annual Conference of the International Speech Communication Association (INTERSPEECH)*. Virtual: ISCA, 2020. arXiv: 2102.09928 (cited on page 150).

[58] Borgholt, L., **Havtorn, J. D.**, Edin, J., Maaløe, L., Igel, C., "A Brief Overview of Neural Speech Representation Learning". In: *Proceedings of the 2nd Workshop on Self-supervised Learning for Audio and Speech Processing (SAS) at the Thirty-Sixth AAAI Conference on Artificial Intelligence*. Virtual, 2022. arXiv: 2203.01829 (cited on pages vii, 13, 79, 178).

[59] Borgholt, L., Tax, T. M. S., **Havtorn, J. D.**, Maaløe, L., Igel, C., "On Scaling Contrastive Representations for Low-Resource Speech Recognition". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Virtual: IEEE, 2021. arXiv: 2102.00850 (cited on pages viii, 83).

[60]   Bourlard, H. A., Morgan, N., *Connectionist Speech Recognition: A Hybrid Approach*. Volume 247. Springer Science & Business Media, 1994 (cited on page 176).

[61]   Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Józefowicz, R., Bengio, S., "Generating Sentences from a Continuous Space". In: *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*. Berlin, Germany, 2016 (cited on pages 92, 101).

[62]   Brandenburg, K., Eberlein, E., Gerhäuser, H., Grill, B., Herre, J., Popp, H., *MPEG-2 Audio Layer III (MP3)*. Germany: Fraunhofer Society, 1998 (cited on page 109).

[63]   Britton, J. P., Proust, C., Shnider, S., "Plimpton 322: A Review and a Different Perspective". In: *Archive for history of exact sciences* 65 (2011) (cited on page 3).

[64]   Brock, A., Donahue, J., Simonyan, K., *Large Scale GAN Training for High Fidelity Natural Image Synthesis*. 2019. arXiv: 1809.11096. (Visited on 22 September 2023) (cited on page 30).

[65]   Brown, M. B. "400: A Method for Combining Non-Independent, One-Sided Tests of Significance". In: *Biometrics. Journal of the International Biometric Society* (1975) (cited on page 64).

[66]   Bu, H., Du, J., Na, X., Wu, B., Zheng, H., "AISHELL-1: An Open-Source Mandarin Speech Corpus and a Speech Recognition Baseline". In: *Proceedings of the 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*. Seoul, South Korea: IEEE, 2017 (cited on page 202).

[67]   Bukhsh, Z. A., Saeed, A., "On Out-of-Distribution Detection for Audio with Deep Nearest Neighbors". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Rhodes Island, Greece: IEEE, 2023 (cited on page 163).

[68]   Bulatov, Y. *notMNIST Dataset*. notMNIST dataset. 2011. URL: http://yaroslavvb.blogspot.com/2011/09/notmnist-dataset.html (cited on page 227).

[69]   Burda, Y., Grosse, R., Salakhutdinov, R. R., "Importance Weighted Autoencoders". In: *Proceedings of the 4th International Conference on Learning Representations (ICLR)*. San Juan, Puerto Rico, 2016 (cited on pages 34–36, 45, 46, 49, 69, 161, 235).

[70]   Burg, G. J. J., Williams, C. K. I., *On Memorization in Probabilistic Deep Generative Models*. 2021. arXiv: 2106.03216 (cited on page 11).

[71]   Burkart, N., Huber, M. F., "A Survey on the Explainability of Supervised Machine Learning". In: *Journal of Artificial Intelligence Research* 70 (2021) (cited on page 11).

[72]   Burns, E., Rigby, E., Mamidanna, R., Bottle, A., Aylin, P., Ziprin, P., Faiz, O., "Systematic Review of Discharge Coding Accuracy". In: *Journal of Public Health (Oxford, England)* 34.1 (2012). ISSN: 1741-3842. pmid: 21795302 (cited on pages 116, 138).

[73]   Burton, D. K., Shore, J. E., Buck, J. T., "A Generalization of Isolated Word Recognition Using Vector Quantization". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (1983) (cited on pages 87, 185).

[74]   Buse, A. "The Likelihood Ratio, Wald, and Lagrange Multiplier Tests: An Expository Note". In: *The American Statistician* 36 (3a 1982) (cited on pages 49, 61).

[75]   Cao, P., Chen, Y., Liu, K., Zhao, J., Liu, S., Chong, W., "HyperCore: Hyperbolic and Co-graph Representation for Automatic ICD Coding". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, 2020 (cited on pages 121, 171).

[76]   Cao, S., Kang, Y., Fu, Y., Xu, X., Sun, S., Zhang, Y., Ma, L., "Improving Streaming Transformer Based ASR under a Framework of Self-Supervised Learning". In: *IEEE Annual Conference of the International Speech Communication Association (Interspeech)*. 2021 (cited on page 225).

[77]   Carcel, C., Woodward, M., Wang, X., Bushnell, C., Sandset, E. C., "Sex Matters in Stroke: A Review of Recent Evidence on the Differences Between Women and Men". In: *Frontiers in Neuroendocrinology* 59 (2020) (cited on page 146).

[78]   Carlin, M. A., Thomas, S., Jansen, A., Hermansky, H., "Rapid Evaluation of Speech Representations for Spoken Term Discovery". In: *Annual Conference of the International Speech Communication Association*. 2011 (cited on page 200).

[79]   Caron, M., Bojanowski, P., Joulin, A., Douze, M., *Deep Clustering for Unsupervised Learning of Visual Features*. 2018. arXiv: 1807.05520 (cited on pages 180, 194).

[80]   Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A., "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2020 (cited on page 180).

[81]   Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A., "Emerging Properties in Self-Supervised Vision Transformers". In: *IEEE International Conference on Computer Vision (ICCV)*. 2021 (cited on page 196).

[82]   Caruana, R. "Multitask Learning". In: *Journal of Machine Learning* 28.1 (1997) (cited on page 181).

[83]   Center for Disease Control and Prevention (CDC), *International Classification of Diseases, (ICD-10-CM/PCS) Transition – Frequently Asked Questions*. 2023. URL: https://www.cdc.gov/nchs/icd/icd10cm_pcs_faq.htm (visited on 2 December 2023) (cited on page 172).

[84]   Chan, D. M., Ghosh, S., "Content-Context Factorized Representations for Automated Speech Recognition". In: *IEEE Annual Conference of the International Speech Communication Association (Interspeech)*. 2022 (cited on page 225).

[85]   Chan, D. M., Ghosh, S., Chakrabarty, D., Hoffmeister, B., "Multi-Modal Pre-Training for Automated Speech Recognition". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022 (cited on page 199).

[86]   Chan, W., Jaitly, N., Le, Q. V., Vinyals, O., "Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016. arXiv: 1508.01211 (cited on page 95).

[87]   Chang, E., Lippmann, R. P., "Figure of Merit Training for Detection and Spotting". In: *Advances in Neural Information Processing Systems* 6 (1993) (cited on page 22).

[88]   Chang, H.-J., Yang, S.-w., Lee, H.-y., "DistilHuBERT: Speech Representation Learning by Layer-wise Distillation of Hidden-unit BERT". 2021. arXiv: 2110.01900 (cited on page 225).

[89]   Chang, K.-W., Tseng, W.-C., Li, S.-W., Lee, H.-y., "SpeechPrompt: An Exploration of Prompt Tuning on Generative Spoken Language Model for Speech Processing Tasks". In: *IEEE Annual Conference of the International Speech Communication Association (Interspeech)*. 2022 (cited on page 224).

[90]   Chang, X., Maekaku, T., Guo, P., Shi, J., Lu, Y.-J., Subramanian, A. S., Wang, T., Yang, S.-w., Tsao, Y., Lee, H.-y., Watanabe, S., "An Exploration of Self-Supervised Pretrained Representations for End-to-End Speech Recognition". 2021. arXiv: 2110.04590 (cited on page 216).

[91]   Chen, G., Parada, C., Sainath, T. N., "Query-by-Example Keyword Spotting Using Long Short-Term Memory Networks". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015 (cited on page 199).

[92]   Chen, G. "GigaSpeech: An Evolving, Multi-Domain ASR Corpus with 10,000 Hours of Transcribed Audio". In: *Annual Conference of the International Speech Communication Association*. 2021 (cited on page 202).

[93] Chen, H., Engkvist, O., Wang, Y., Olivecrona, M., Blaschke, T., "The Rise of Deep Learning in Drug Discovery". In: *Drug discovery today* 23.6 (2018) (cited on page 5).

[94] Chen, I. Y., Pierson, E., Rose, S., Joshi, S., Ferryman, K., Ghassemi, M., "Ethical Machine Learning in Healthcare". In: *Annual review of biomedical data science* 4 (2021) (cited on page 3).

[95] Chen, J., Sathe, S., Aggarwal, C., Turaga, D., "Outlier Detection with Autoencoder Ensembles". In: *Proceedings of the SIAM International Conference on Data Mining (SDM)*. SIAM, 2017 (cited on pages 24, 29).

[96] Chen, K.-Y., Tsai, C.-P., Liu, D.-R., Lee, H.-Y., Lee, L.-s., "Completely Unsupervised Speech Recognition by A Generative Adversarial Network Harmonized with Iteratively Refined Hidden Markov Models". In: *Proceedings of the 20th Annual Conference of the International Speech Communication Association (Interspeech)*. ISCA, 2019 (cited on pages 218, 219).

[97] Chen, S., Wu, Y., Wang, C., Chen, Z., Chen, Z., Liu, S., Wu, J., Qian, Y., Wei, F., Li, J., Yu, X., *UniSpeech-SAT: Universal Speech Representation Learning with Speaker Aware Pre-Training*. 2021. arXiv: 2110.05752 (cited on pages 202, 205, 207–209, 211).

[98] Chen, S. "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing". 2021. arXiv: 2110.13900 (cited on pages 95, 190, 195, 202, 205, 207–209, 211, 214, 215).

[99] Chen, T., Kornblith, S., Norouzi, M., Hinton, G., "A Simple Framework for Contrastive Learning of Visual Representations". In: *Proceedings of the 37th International Conference on Machine Learning (ICML)*. 2020 (cited on pages 28, 38, 180).

[100] Chen, X., Fan, H., Girshick, R., He, K., *Improved Baselines with Momentum Contrastive Learning*. 2020. arXiv: 2003.04297 (cited on page 180).

[101] Chi, P.-H., Chung, P.-H., Wu, T.-H., Hsieh, C.-C., Chen, Y.-H., Li, S.-W., Lee, H.-y., "Audio ALBERT: A Lite BERT for Self-supervised Learning of Audio Representation". In: (2020) (cited on pages 83, 190, 206).

[102] Child, R. "Very Deep VAEs Generalize Autoregressive Models and Can Outperform Them on Images". In: *Proceedings of the 9th International Conference on Learning Representations (ICLR)*. 2021 (cited on pages 30, 37, 43, 46, 47, 55, 97, 99, 105, 159–162, 170, 228, 235, 250).

[103] Chira, D., Haralampiev, I., Winther, O., Dittadi, A., Liévin, V., *Image Super-Resolution With Deep Variational Autoencoders*. 2022. arXiv: 2203.09445. (Visited on 25 September 2023) (cited on page 38).

[104] Chiu, C.-C., Qin, J., Zhang, Y., Yu, J., Wu, Y., *Self-Supervised Learning with Random-projection Quantizer for Speech Recognition*. 2022. arXiv: 2202.01855 (cited on page 190).

[105] Cho, K., Merrienboer, B., Bahdanau, D., Bengio, Y., *On the Properties of Neural Machine Translation: Encoder-decoder Approaches*. 2014. arXiv: 1409.1259 (cited on pages 101, 252).

[106] Choi, H.-S., Lee, J., Kim, W., Lee, J., Heo, H., Lee, K., "Neural Analysis and Synthesis: Reconstructing Speech from Self-Supervised Representations". In: *Conference on Neural Information Processing Systems (NeurIPS)*. Virtual, 2021 (cited on page 225).

[107] Choi, H., Jang, E., Alemi, A. A., *WAIC, but Why? Generative Ensembles for Robust Anomaly Detection*. 2019. arXiv: 1810.01392 (cited on pages 24, 28, 42, 47, 51, 53, 55, 157, 163).

[108] Choi, J., Yoon, C., Bae, J., Kang, M., "Robust Out-of-Distribution Detection on Deep Probabilistic Generative Models". 2021. arXiv: 2106.07903 (cited on page 69).

[109] Chorowski, J., Jaitly, N., "Towards Better Decoding and Language Model Integration in Sequence to Sequence Models". In: *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*. ISCA, 2017 (cited on page 221).

[110]   Chorowski, J., Weiss, R. J., Bengio, S., Oord, A., "Unsupervised Speech Representation Learning Using WaveNet Autoencoders". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.12 (2019). ISSN: 2329-9290, 2329-9304. arXiv: 1901.08810 (cited on pages 87, 89, 94, 95, 186, 223).

[111]   Chrupała, G. *Visually Grounded Models of Spoken Language: A Survey of Datasets, Architectures and Evaluation Techniques*. 2021. arXiv: 2104.13225 (cited on page 198).

[112]   Chrupała, G., Gelderloos, L., Alishahi, A., "Representations of Language in a Model of Visually Grounded Speech Signal". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2017 (cited on page 199).

[113]   Chung, Y.-A., Belinkov, Y., Glass, J., "Similarity Analysis of Self-Supervised Speech Representations". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021 (cited on page 214).

[114]   Chung, Y.-A., Glass, J., "Generative Pre-Training for Speech with Autoregressive Predictive Coding". In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020 (cited on pages 81, 95, 186, 206).

[115]   Chung, Y.-A., Glass, J., *Improved Speech Representations with Multi-Target Autoregressive Predictive Coding*. 2020. arXiv: 2004.05274 (cited on pages 83, 186, 206).

[116]   Chung, Y.-A., Glass, J., "Learning Word Embeddings from Speech". 2017. arXiv: 1711.01515 (cited on page 88).

[117]   Chung, Y.-A., Glass, J., "Speech2Vec: A Sequence-to-Sequence Framework for Learning Word Embeddings from Speech". 2018. arXiv: 1803.08976 (cited on pages 85, 88, 94, 188, 190, 199, 218, 219).

[118]   Chung, Y.-A., Hsu, W.-N., Tang, H., Glass, J., *An Unsupervised Autoregressive Model for Speech Representation Learning*. 2019. arXiv: 1904.03240 (cited on pages 81, 82, 85, 95, 186, 190, 206).

[119]   Chung, Y.-A., Tang, H., Glass, J., "Vector-Quantized Autoregressive Predictive Coding". 2020. arXiv: 2005.08392 (cited on pages 87, 186, 190, 207).

[120]   Chung, Y.-A., Weng, W.-H., Tong, S., Glass, J., "Towards Unsupervised Speech-to-Text Translation". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019 (cited on pages 218, 219).

[121]   Chung, Y.-A., Weng, W.-H., Tong, S., Glass, J., "Unsupervised Cross-Modal Alignment of Speech and Text Embedding Spaces". In: *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS)* (2018) (cited on pages 218, 219).

[122]   Chung, Y.-A., Wu, C.-C., Shen, C.-H., Lee, H.-Y., Lee, L.-S., "Audio Word2Vec: Unsupervised Learning of Audio Segment Representations Using Sequence-to-sequence Autoencoder". In: (2016) (cited on pages 85, 88, 94, 190, 199, 200).

[123]   Chung, Y.-A., Zhang, Y., Han, W., Chiu, C.-C., Qin, J., Pang, R., Wu, Y., "W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training". In: (2021) (cited on pages 190, 195).

[124]   Chung, Y.-A., Zhu, C., Zeng, M., *SPLAT: Speech-Language Joint Pre-Training for Spoken Language Understanding*. 2021. arXiv: 2010.02295 (cited on page 95).

[125]   Chung, J. S., Zisserman, A., "Lip Reading in the Wild". In: *Asian Conference on Computer Vision*. 2016 (cited on page 198).

[126]   Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., Bengio, Y., "A Recurrent Latent Variable Model for Sequential Data". In: *Proceedings of the 29th Conference on Neural Information Processing Systems (NeurIPS)*. Montréal, Quebec, Canada, 2015 (cited on pages 85, 91–93, 97–100, 106, 185, 242, 243, 246, 249, 251, 255).

[127] Cieri, C., Miller, D., Walker, K., "The Fisher Corpus: A Resource for the next Generations of Speech-to-Text". In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*. Volume 4. 2004 (cited on page 201).

[128] Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., Ha, D., *Deep Learning for Classical Japanese Literature*. 2018. arXiv: 1812.01718 (cited on page 227).

[129] Clark, K., Luong, M.-T., Le, Q. V., Manning, C. D., *ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators*. 2020. arXiv: 2003.10555. (Visited on 23 September 2023) (cited on page 96).

[130] Coalson, J., Castro Lopo, E., *Free Lossless Audio Encoding (FLAC)*. Version 1.3.3. xiph.org, 2019 (cited on pages 106, 109).

[131] Conneau, A., Baevski, A., Collobert, R., Mohamed, A., Auli, M., *Unsupervised Cross-lingual Representation Learning for Speech Recognition*. 2020. arXiv: 2006.13979. (Visited on 29 October 2021) (cited on pages 89, 216).

[132] Conneau, A., Lample, G., Ranzato, M., Denoyer, L., Jégou, H., "Word Translation without Parallel Data". In: *International Conference on Learning Representations (ICLR)*. 2018 (cited on pages 218, 219).

[133] Cook, M., Zare, A., Gader, P., *Outlier Detection through Null Space Analysis of Neural Networks*. 2020. arXiv: 2007.01263. (Visited on 17 September 2023) (cited on pages 24, 27).

[134] Cortes, C., Vapnik, V., "Support-Vector Networks". In: *Journal of Machine Learning* 20.3 (1995) (cited on page 28).

[135] Cover, T., Hart, P., "Nearest Neighbor Pattern Classification". In: *IEEE Transactions on Information Theory* 13.1 (1967) (cited on page 28).

[136] Cremer, C., Li, X., Duvenaud, D., "Inference Suboptimality in Variational Autoencoders". In: *Proceedings of the 35th International Conference on Machine Learning (ICML)*. Volume 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm, Sweden: PMLR, 2018 (cited on pages 33, 47, 62).

[137] Cuervo, S., Grabias, M., Chorowski, J., Ciesielski, G., Łańcucki, A., Rychlikowski, P., Marxer, R., "Contrastive Prediction Strategies for Unsupervised Segmentation and Categorization of Phonemes and Words". In: *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022 (cited on page 89).

[138] Cui, J. "Multilingual Representations for Low Resource Speech Recognition and Keyword Search". In: *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)* (2015) (cited on page 181).

[139] Cuneiform Digital Library Initiative (CDLI), *MCT 038, Plimpton 322 artifact entry*. 2005 (cited on page 3).

[140] Cuneiform Digital Library Initiative (CDLI), *MSVO 3, 02 (P005313)*. 2018 (cited on page 4).

[141] D'Angelo, F., Henning, C., *On Out-of-Distribution Detection with Bayesian Neural Networks*. 2022. arXiv: 2110.06020. (Visited on 19 September 2023) (cited on page 27).

[142] Dangel, F., Kunstner, F., Hennig, P., "BackPACK: Packing More into Backprop". In: *International Conference on Learning Representations (ICLR)*. 2020 (cited on page 66).

[143] Danmarks Statistik (Statistics Denmark), *FOLK1: Population Quarterly Database (February 2023)*. Danmarks Statistik (Statistics Denmark), 2023 (cited on page 148).

[144] Danske Regioner, Laerdal, *Dansk Indeks for Akuthjælp. Landsudgaven, Version 1.10 (Danish Index for Emergency Care. Nation Edition, Version 1.10)*. 2022 (cited on page 149).

[145] Dehak, N., Torres-Carrasquillo, P., Reynolds, D., Dehak, R., "Language Recognition via I-Vectors and Dimensionality Reduction". In: *Annual Conference of the International Speech Communication Association*. 2011 (cited on page 179).

[146] Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., Ouellet, P., "Front-End Factor Analysis for Speaker Verification". In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.4 (2011). ISSN: 1558-7924 (cited on page 179).

[147] Dempster, A. P., Laird, N. M., Rubin, D. B., "Maximum Likelihood from Incomplete Data Via the Em Algorithm". In: *Journal of the Royal Statistical Society: Series B (methodological)* 39.1 (1977) (cited on pages 28, 31).

[148] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., "ImageNet: A Large-Scale Hierarchical Image Database". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009. ISBN: 978-1-4244-3992-8. pmid: 21914436 (cited on page 79).

[149] Deng, L., Seltzer, M., Yu, D., Acero, A., Mohamed, A., Hinton, G., *Binary Coding of Speech Spectrograms Using a Deep Auto-encoder*. 2010 (cited on page 94).

[150] Denouden, T., Salay, R., Czarnecki, K., Abdelzad, V., Phan, B., Vernekar, S., *Improving Reconstruction Autoencoder Out-of-distribution Detection with Mahalanobis Distance*. 2018. arXiv: 1812.02765. (Visited on 10 September 2023) (cited on pages 24, 29, 30).

[151] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2018. arXiv: 1810.04805. (Visited on 11 February 2019) (cited on pages 28, 38, 79, 83, 87, 94, 96, 137, 162, 171, 180, 187, 194, 226).

[152] DeVries, T., Taylor, G. W., *Learning Confidence for Out-of-Distribution Detection in Neural Networks*. 2018. arXiv: 1802.04865 (cited on pages 25, 41).

[153] Dhamija, A. R., Günther, M., Boult, T., "Reducing Network Agnostophobia". In: *Advances in Neural Information Processing Systems* 31 (2018) (cited on page 24).

[154] Dieleman, S., Nash, C., Engel, J., Simonyan, K., *Variable-Rate Discrete Representation Learning*. 2021. arXiv: 2103.06089 (cited on page 89).

[155] Dieng, A. B., Kim, Y., Rush, A. M., Blei, D. M., "Avoiding Latent Variable Collapse with Generative Skip Models". In: *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*. Volume 89. Naha, Okinawa, Japan: PMLR, 2019 (cited on pages 50, 51).

[156] Dinh, L., Krueger, D., Bengio, Y., "NICE: Non-linear Independent Components Estimation". In: *3rd International Conference on Learning Representations Workshop*. San Diego, CA, USA, 2015. arXiv: 1410.8516 (cited on pages 28, 30, 105).

[157] Dinh, L., Sohl-Dickstein, J., Bengio, S., "Density Estimation Using Real NVP". In: *5th International Conference on Learning Representations*. Palais des Congrès Neptune, Toulon, France, 2017. arXiv: 1605.08803 (cited on page 30).

[158] Doersch, C., Gupta, A., Efros, A. A., "Unsupervised Visual Representation Learning by Context Prediction". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015. arXiv: 1505.05192 (cited on pages 79, 180).

[159] Dong, H., Falis, M., Whiteley, W., Alex, B., Matterson, J., Ji, S., Chen, J., Wu, H., "Automated Clinical Coding: What, Why, and Where We Are?" In: *npj Digital Medicine* 5.1 (2022). ISSN: 2398-6352 (cited on page 116).

[160] Dong, H., Suárez-Paniagua, V., Whiteley, W., Wu, H., "Explainable Automated Coding of Clinical Notes Using Hierarchical Label-Wise Attention Networks and Label Embedding Initialisation". In: *Journal of Biomedical Informatics* 116 (2021). ISSN: 1532-0480. pmid: 33711543 (cited on pages 116, 137).

[161] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., "An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. Virtual, 2021. arXiv: 2010.11929 (cited on pages 137, 171).

[162] Du, Y., Mordatch, I., "Implicit Generation and Modeling with Energy Based Models". In: *Advances in Neural Information Processing Systems* 32 (2019) (cited on page 30).

[163] Dunbar, E., Algayres, R., Karadayi, J., Bernard, M., Benjumea, J., Cao, X.-N., Miskic, L., Dugrain, C., Ondel, L., Black, A. W., *The Zero Resource Speech Challenge 2019: TTS Without T*. 2019. arXiv: 1904.11469 (cited on pages 95, 222).

[164] Dunbar, E., Bernard, M., Hamilakis, N., Nguyen, T., Seyssel, M., Rozé, P., Rivière, M., Kharitonov, E., Dupoux, E., "The Zero Resource Speech Challenge 2021: Spoken Language Modelling". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021) (cited on pages 95, 157).

[165] Dunbar, E., Cao, X. N., Benjumea, J., Karadayi, J., Bernard, M., Besacier, L., Anguera, X., Dupoux, E., "The Zero Resource Speech Challenge 2017". In: *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. 2017 (cited on pages 95, 222).

[166] Dunbar, E., Karadayi, J., Bernard, M., Cao, X.-N., Algayres, R., Ondel, L., Besacier, L., Sakti, S., Dupoux, E., "The Zero Resource Speech Challenge 2020: Discovering Discrete Subword and Word Units". In: *Annual Conference of the International Speech Communication Association*. 2020 (cited on pages 95, 211, 212, 222).

[167] Dwass, M. "Modified Randomization Tests for Nonparametric Hypotheses". In: *The Annals of Mathematical Statistics* 28.1 (1957) (cited on pages 152, 281).

[168] Ebbers, J., Heymann, J., Drude, L., Glarner, T., Haeb-Umbach, R., Raj, B., "Hidden Markov Model Variational Autoencoder for Acoustic Unit Discovery". In: *Proceedings of the 18th Annual Conference of the International Speech Communication Association (Interspeech)*. ISCA, 2017 (cited on pages 85, 92).

[169] Eddelien, H. S., Butt, J. H., Christensen, T., Danielsen, A. K., Kruuse, C., "Sex and Age Differences in Patient-Reported Acute Stroke Symptoms". In: *Frontiers in Neurology* 13 (2022) (cited on page 146).

[170] Eden, T., Yates, F., "On the Validity of Fisher's z Test When Applied to an Actual Example of Non-Normal Data. (With Five Text-Figures.)" In: *The Journal of Agricultural Science* 23.1 (1933) (cited on pages 152, 281).

[171] Edin, J., Junge, A., **Havtorn, J. D.**, Borgholt, L., Maistro, M., Ruotsalo, T., Maaløe, L., "Automated Medical Coding on MIMIC-III and MIMIC-IV: A Critical Review and Replicability Study". In: *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. Taipei, Taiwan: ACM, 2023. arXiv: 2304.10909 (cited on pages vii, 13, 115).

[172] Eloff, R., Nortje, A., Niekerk, B., Govender, A., Nortje, L., Pretorius, A., Biljon, E., Westhuizen, E., Staden, L., Kamper, H., "Unsupervised Acoustic Unit Discovery for Speech Synthesis Using Discrete Latent-Variable Neural Networks". 2019. arXiv: 1904.07556 (cited on page 186).

[173] Embi, P. J., Leonard, A. C., "Evaluating Alert Fatigue over Time to EHR-based Clinical Trial Alerts: Findings from a Randomized Controlled Study". In: *Journal of the American Medical Informatics Association* 19.e1 (2012). ISSN: 1067-5027 (cited on page 166).

[174] Engel, J., Resnick, C., Roberts, A., Dieleman, S., Norouzi, M., Eck, D., Simonyan, K., "Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders". In: *Proceedings of the International Conference on Machine Learning (ICML)*. 2017 (cited on pages 204, 205).

[175]   Ephrat, A., Mosseri, I., Lang, O., Dekel, T., Wilson, K., Hassidim, A., Freeman, W. T., Ru-
        binstein, M., "Looking to Listen at the Cocktail Party: A Speaker-Independent Audio-Visual
        Model for Speech Separation". In: *ACM Transactions on Graphics (TOG)* 37.4 (2018) (cited on
        page 201).

[176]   Ericsson, L., Gouk, H., Loy, C. C., Hospedales, T. M., "Self-Supervised Representation Learn-
        ing: Introduction, Advances, and Challenges". In: *IEEE Signal Processing Magazine* 39.3 (2022)
        (cited on page 178).

[177]   European Commission, *Briefing on the Artificial Intelligence Act*. 2021 (cited on page 7).

[178]   European Parliament, Directorate-General for Parliamentary Research Services, Lekadir, K.,
        Quaglio, G., Tselioudis Garmendia, A., Gallin, C., *Artificial Intelligence in Healthcare – Applica-
        tions, Risks, and Ethical and Societal Impacts*. European Parliament, 2022 (cited on page 7).

[179]   Evain, S. "LeBenchmark: A Reproducible Framework for Assessing Self-Supervised Repre-
        sentation Learning from Speech". In: *Annual Conference of the International Speech Communica-
        tion Association*. 2021 (cited on pages 211, 212).

[180]   Faruqi, M., Dyer, C., "Community Evaluation and Exchange of Word Vectors and Wordvec-
        tors.Org". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Lin-
        guistics (ACL)*. 2014 (cited on page 213).

[181]   Feucht, M., Wu, Z., Althammer, S., Tresp, V., "Description-Based Label Attention Classifier
        for Explainable ICD-9 Classification". In: *Proceedings of the Seventh Workshop on Noisy User-
        generated Text (W-NUT 2021)*. Online: Association for Computational Linguistics, 2021 (cited
        on pages 116, 137).

[182]   Fischer, A., Igel, C., "Bounding the Bias of Contrastive Divergence Learning". In: *Neural Com-
        putation* 23 (2011) (cited on page 94).

[183]   Fischer, A., Igel, C., "Training Restricted Boltzmann Machines: An Introduction". In: *Pattern
        Recognition* 47 (2014) (cited on page 94).

[184]   Fisher, R. *Statistical Methods for Research Workers*. Edinburgh Oliver & Boyd, 1925 (cited on
        pages 29, 64, 65).

[185]   Fisher, R. A. "The Use of Multiple Measurements in Taxonomic Problems". In: *Annals of eu-
        genics* 7.2 (1936) (cited on page 110).

[186]   Folks, J., Little, R., "Asymptotic Optimality of Fisher's Method of Combining Independent
        Tests". In: *Journal of the American Statistical Association* (1971) (cited on pages 64, 65).

[187]   Fowler, D., Robson, E., "Square Root Approximations in Old Babylonian Mathematics: YBC
        7289 in Context". In: *Historia mathematica* 25.4 (1998) (cited on page 3).

[188]   Fraccaro, M., Sønderby, S. K., Paquet, U., Winther, O., "Sequential Neural Models with Stochas-
        tic Layers". In: *Proceedings of the 30th Conference on Neural Information Processing Systems (NeurIPS)*.
        Barcelona, Spain, 2016 (cited on pages 85, 91–93, 97–99, 101, 106, 185, 242, 243, 246, 249, 251,
        256).

[189]   Frellsen, J., Mattei, P.-A., "Deep Latent Variable Models: Estimation and Missing Data Impu-
        tation" (University of Copenhagen, Datalogisk Institut). 2019 (cited on page 33).

[190]   Fukushima, K. "Neocognitron: A Self-organizing Neural Network Model for a Mechanism
        of Pattern Recognition Unaffected by Shift in Position". In: *Biological Cybernetics* 36.4 (1980)
        (cited on page 228).

[191]   Gal, Y., Ghahramani, Z., *Dropout as a Bayesian Approximation: Representing Model Uncertainty
        in Deep Learning*. 2016. arXiv: 1506.02142 (cited on page 13).

[192] Gales, M. J., Knill, K. M., Ragni, A., Rath, S. P., "Speech Recognition and Keyword Spotting for Low-Resource Languages: BABEL Project Research at CUED". In: *Fourth International Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU)*. 2014 (cited on page 202).

[193] Galke, L., Scherp, A., *Bag-of-Words vs. Graph vs. Sequence in Text Classification: Questioning the Necessity of Text-Graphs and the Surprising Strength of a Wide MLP*. 2022. arXiv: 2109.03777. (Visited on 22 September 2023) (cited on page 147).

[194] Gao, S., Alawad, M., Young, M. T., Gounley, J., Schaefferkoetter, N., Yoon, H. J., Wu, X.-C., Durbin, E. B., Doherty, J., Stroup, A., Coyle, L., Tourassi, G., "Limitations of Transformers on Clinical Text Classification". In: *IEEE journal of biomedical and health informatics* 25.9 (2021). ISSN: 2168-2208. pmid: 33635801 (cited on pages 116, 137, 171).

[195] Garczarek, U. "Classification Rules in Standardized Partition Spaces". In: (2002) (cited on page 11).

[196] Garofolo, J. S. *TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1*. Linguistic Data Consortium, 1993 (cited on pages 82, 105, 163, 186, 202, 242).

[197] Gauthier, E., Besacier, L., Voisin, S., Melese, M., Elingui, U. P., "Collecting Resources in Sub-Saharan African Languages for Automatic Speech Recognition: A Case Study of Wolof". In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*. 2016 (cited on page 204).

[198] Gauvain, J.-L., Lee, C.-H., "Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains". In: *IEEE Transactions on Speech and Audio Processing* 2.2 (1994) (cited on page 178).

[199] GBD 2019 Stroke Collaborators, "Global, Regional, and National Burden of Stroke and Its Risk Factors, 1990–2019: A Systematic Analysis for the Global Burden of Disease Study 2019". In: *The Lancet Neurology* 20.10 (2021). ISSN: 1474-4422 (cited on pages 8, 139).

[200] Geifman, Y., El-Yaniv, R., "Selective Classification for Deep Neural Networks". In: *Advances in neural information processing systems* 30 (2017) (cited on page 171).

[201] Gelas, H., Besacier, L., Pellegrino, F., "Developments of Swahili Resources for an Automatic Speech Recognition System". In: *Spoken Language Technologies for Under-Resourced Languages*. 2012 (cited on page 204).

[202] Gemmeke, J. F., Ellis, D. P., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., Ritter, M., "Audio Set: An Ontology and Human-Labeled Dataset for Audio Events". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017 (cited on page 200).

[203] Georges, I. *The Universal History of Computing: From the Abacus to the Quantum Computer*. Wiley, New York, 2001. ISBN: 978-0-471-39671-0 (cited on pages 3, 5).

[204] Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M. W., Keutzer, K., *A Survey of Quantization Methods for Efficient Neural Network Inference*. 2021. arXiv: 2103.13630. (Visited on 31 July 2022) (cited on page 225).

[205] Girin, L., Leglaive, S., Bie, X., Diard, J., Hueber, T., Alameda-Pineda, X., "Dynamical Variational Autoencoders: A Comprehensive Review". In: *Foundations and Trends in Machine Learning* 15 (2021) (cited on page 180).

[206] Girshick, R. "Fast R-Cnn". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2015 (cited on page 197).

[207] Glarner, T., Hanebrink, P., Ebbers, J., Haeb-Umbach, R., "Full Bayesian Hidden Markov Model Variational Autoencoder for Acoustic Unit Discovery". In: *Proceedings of the 19th Annual Conference of the International Speech Communication Association (Interspeech)*. Hyderabad, India: ISCA, 2018 (cited on pages 85, 92).

[208] Godfrey, J. J., Holliman, E. C., McDaniel, J., "SWITCHBOARD: Telephone Speech Corpus for Research and Development". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 1992 (cited on page 202).

[209] Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C. K., Stanley, H. E., "PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals". In: *Circulation* 101.23 (2000). ISSN: 1524-4539. pmid: 10851218 (cited on pages 116, 138).

[210] Goodfellow, I. "Efficient Per-Example Gradient Computations". 2015. arXiv: 1510.01799 (cited on page 66).

[211] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., "Generative Adversarial Networks". In: *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)* (2014). ISSN: 10495258. pmid: 1000183096 (cited on pages 28, 30, 217).

[212] Goodfellow, I. J., Shlens, J., Szegedy, C., "Explaining and Harnessing Adversarial Examples". In: *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. San Diego, CA, USA, 2015 (cited on page 41).

[213] Govindarajan, P., Desouza, N. T., Pierog, J., Ghilarducci, D., Johnston, S. C., "Feasibility Study to Assess the Use of the Cincinnati Stroke Scale by Emergency Medical Dispatchers: A Pilot Study". In: *Emergency Medicine Journal* 29.10 (2012) (cited on page 140).

[214] Goyal, A., Sordoni, A., Côté, M.-A., Ke, N. R., Bengio, Y., "Z-Forcing: Training Stochastic Recurrent Networks". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*. Long Beach, CA, USA, 2017. arXiv: 1711.05411 (cited on page 99).

[215] Graham, M. S., Pinaya, W. H., Tudosiu, P.-D., Nachev, P., Ourselin, S., Cardoso, J., "Denoising Diffusion Models for Out-of-Distribution Detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023 (cited on pages 23, 24, 29).

[216] Grathwohl, W., Chen, R. T. Q., Bettencourt, J., Sutskever, I., Duvenaud, D., *FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models*. 2018. arXiv: 1810.01367 (cited on page 30).

[217] Graves, A., Fernández, S., Gomez, F., Schmidhuber, J., "Connectionist Temporal Classification". In: *Proceedings of the International Conference on Machine Learning (ICML)*. Pittsburgh, Pennsylvania, USA, 2006. ISBN: 1-59593-383-2. pmid: 1000285842 (cited on pages 95, 110, 150, 151).

[218] Gray, R. "Vector Quantization". In: *IEEE ASSP Magazine* 1.2 (1984) (cited on page 177).

[219] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., Smola, A., "A Kernel Two-Sample Test". In: *The Journal of Machine Learning Research* 13.1 (2012) (cited on page 62).

[220] Grigg*, T. G., Busbridge*, D., Ramapuram, J., Webb, R., *Do Self-Supervised and Supervised Methods Learn Similar Visual Representations?* 2021 (cited on page 213).

[221] Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. Á., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M., *Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning*. 2020. arXiv: 2006.07733 (cited on page 196).

[222] Guidi, J. L. "Clinician Perception of the Effectiveness of an Automated Early Warning and Response System for Sepsis in an Academic Medical Center". In: *Annals of the American Thoracic Society* 12.10 (2015). ISSN: 2325-6621. PMID: 26288388 (cited on page 166).

[223] Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., Pang, R., *Conformer: Convolution-augmented Transformer for Speech Recognition*. 2020. arXiv: 2005.08100. (Visited on 17 March 2023) (cited on page 95).

[224] Gulcehre, C., Firat, O., Xu, K., Cho, K., Barrault, L., Lin, H.-C., Bougares, F., Schwenk, H., Bengio, Y., *On Using Monolingual Corpora in Neural Machine Translation*. 2015. arXiv: 1503.03535. (Visited on 12 April 2023) (cited on page 221).

[225] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A., "Improved Training of Wasserstein GANs". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*. 2017 (cited on pages 217–219).

[226] Guo, C., Pleiss, G., Sun, Y., Weinberger, K. Q., *On Calibration of Modern Neural Networks*. 2017. arXiv: 1706.04599 (cited on pages 11, 25).

[227] Guo, D., Rush, A., Kim, Y., "Parameter-Efficient Transfer Learning with Diff Pruning". In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. Virtual: Association for Computational Linguistics, 2021 (cited on page 224).

[228] Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., Smith, N. A., "Don't Stop Pretraining: Adapt Language Models to Domains and Tasks". 2020. arXiv: 2004.10964 (cited on page 147).

[229] Gutmann, M. U., Hyvärinen, A., "Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics." In: *Journal of Machine Learning Research* 13.2 (2010) (cited on pages 82, 179, 191).

[230] Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., Davidson, J., "Learning Latent Dynamics for Planning from Pixels". In: *Proceedings of the 36th International Conference on Machine Learning*. Volume 97. Proceedings of Machine Learning Research. PMLR, 2019 (cited on page 251).

[231] Hajavi, A., Etemad, A., "Siamese Capsule Network for End-to-End Speaker Recognition in the Wild". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021 (cited on page 211).

[232] Hajibabaei, M., Dai, D., *Unified Hypersphere Embedding for Speaker Recognition*. 2018. arXiv: 1807.08312 (cited on page 211).

[233] Han, K. J., Hahm, S., Kim, B.-H., Kim, J., Lane, I. R., "Deep Learning-Based Telephony Speech Recognition in the Wild." In: *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*. 2017 (cited on page 146).

[234] Hansen, L. K., Salamon, P., "Neural Network Ensembles". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.10 (1990) (cited on page 150).

[235] Harari, Y. N. *Sapiens: A Brief History of Humankind*. 2011. ISBN: 978-0-06-231609-7 (cited on page 5).

[236] Hariharan, P., Tariq, M. B., Grotta, J. C., Czap, A. L., "Mobile Stroke Units: Current Evidence and Impact". In: *Current Neurology and Neuroscience Reports* 22.1 (2022) (cited on pages 8, 140).

[237] Haroush, M., Frostig, T., Heller, R., Soudry, D., "A Statistical Framework for Efficient out of Distribution Detection in Deep Neural Networks". 2021. arXiv: 2102.12967 (cited on pages 60, 69).

[238] Harwath, D., Chuang, G., Glass, J., "Vision as an Interlingua: Learning Multilingual Semantic Embeddings of Untranscribed Speech". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018 (cited on page 199).

[239] Harwath, D., Glass, J. R., "Deep Multimodal Semantic Embeddings for Speech and Images". In: *Ieee Workshop on Automatic Speech Recognition and Understanding*. 2015 (cited on page 199).

[240] Harwath, D., Hsu, W.-N., Glass, J., "Learning Hierarchical Discrete Linguistic Units from Visually-Grounded Speech". In: *International Conference on Learning Representations (ICLR)*. 2019 (cited on page 199).

[241] Harwath, D., Recasens, A., Surís, D., Chuang, G., Torralba, A., Glass, J., "Jointly Discovering Visual Objects and Spoken Words from Raw Sensory Input". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018 (cited on page 199).

[242] Harwath, D., Torralba, A., Glass, J. R., "Unsupervised Learning of Spoken Language with Visual Context". In: *Conference on Neural Information Processing Systems (Neurips)*. 2016 (cited on page 199).

[243] Havard, W. N., Chevrot, J.-P., Besacier, L., "Models of Visually Grounded Speech Signal Pay Attention to Nouns: A Bilingual Experiment on English and Japanese". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019 (cited on page 199).

[244] **Havtorn, J. D.**, Borgholt, L., Hauberg, S., Frellsen, J., Maaløe, L., "Benchmarking Generative Latent Variable Models for Speech". In: *Proceedings of the Workshop on Deep Generative Models for Highly Structured Data at ICML*. 2022. arXiv: 2202.12707 (cited on pages vii, 13, 97).

[245] **Havtorn, J. D.**, Frellsen, J., Hauberg, S., Maaløe, L., "Hierarchical VAEs Know What They Don't Know". In: *Proceedings of the 38th International Conference on Machine Learning (ICML)*. Virtual: PMLR, 2021. arXiv: 2102.08248 (cited on pages vii, 13, 24, 28, 41, 69, 98, 263, 264).

[246] **Havtorn, J. D.**, Latko, J., Edin, J., Borgholt, L., Maaløe, L., Belgrano, L., Jakobsen, N. F., Sdun, R., Agić, Ž., "MultiQT: Multimodal Learning for Real-Time Question Tracking in Speech". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*. Virtual: Association for Computational Linguistics, 2020. arXiv: 2005.00812 (cited on page 8).

[247] Hayashi, T., Watanabe, S., *DiscreTalk: Text-to-Speech as a Machine Translation Problem*. 2020. arXiv: 2005.05525. (Visited on 12 April 2023) (cited on page 224).

[248] Hayashi, T., Watanabe, S., Zhang, Y., Toda, T., Hori, T., Astudillo, R., Takeda, K., "Back-Translation-Style Data Augmentation for End-to-End ASR". In: *IEEE Spoken Language Technology Workshop*. 2018 (cited on page 221).

[249] He, K., Fan, H., Wu, Y., Xie, S., Girshick, R., "Momentum Contrast for Unsupervised Visual Representation Learning". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020 (cited on page 180).

[250] He, K., Zhang, X., Ren, S., Sun, J., "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2015). arXiv: 1512.03385 (cited on page 79).

[251] Heck, M., Sakti, S., Nakamura, S., "Feature Optimized DPGMM Clustering for Unsupervised Subword Modeling: A Contribution to Zerospeech 2017". In: *2017 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2017, Okinawa, Japan, December 16-20, 2017*. 2017 (cited on pages 94, 223).

[252] Hendrycks, D., Basart, S., Mazeika, M., Zou, A., Kwon, J., Mostajabi, M., Steinhardt, J., Song, D., "Scaling Out-of-Distribution Detection for Real-World Settings". In: *Proceedings of the International Conference on Machine Learning (ICML)*. Volume 162. Proceedings of Machine Learning Research. Baltimore, Maryland, USA: PMLR, 2022. arXiv: 1911.11132 (cited on pages 24, 26).

[253] Hendrycks, D., Gimpel, K., "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks". In: *Proceedings of the 5th International Conference on Learning Representations (ICRL)*. Toulon, France, 2017 (cited on pages 23–25, 41, 47, 51, 53, 163, 164).

[254] Hendrycks, D., Mazeika, M., Dietterich, T. G., "Deep Anomaly Detection with Outlier Exposure". In: *Proceedings of the 7th International Conference on Learning Representations (ICLR)*. New Orleans, LA, USA, 2019 (cited on pages 24, 26, 28, 41, 47, 50, 51, 53, 60, 68, 69, 72, 157, 163, 271–273).

[255] Hendrycks, D., Mazeika, M., Kadavath, S., Song, D., "Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty". In: *Advances in Neural Information Processing Systems* 32 (2019) (cited on pages 24, 30, 157).

[256] Henning, C., D'Angelo, F., Grewe, B. F., "Are Bayesian Neural Networks Intrinsically Good at Out-of-Distribution Detection?" In: *Proceedings of the ICML 2021 Workshop on Uncertainty & Robustness in Deep Learning*. Virtual, 2021 (cited on page 27).

[257] Hermann, K. M., Blunsom, P., "Multilingual Distributed Representations Without Word Alignment". In: *International Conference on Learning Representations (ICLR)*. 2013. arXiv: 1312.6173 (cited on page 198).

[258] Hernandez, F., Nguyen, V., Ghannay, S., Tomashenko, N., Esteve, Y., "TED-LIUM 3: Twice as Much Data and Corpus Repartition for Experiments on Speaker Adaptation". In: *International Conference on Speech and Computer*. 2018 (cited on page 202).

[259] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A., "β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. Palais des Congrès Neptune, Toulon, France, 2017 (cited on page 32).

[260] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., Kingsbury, B., "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups". In: *IEEE Signal Processing Magazine* 29.6 (2012) (cited on page 176).

[261] Hinton, G. E. "Learning Multiple Layers of Representation". In: *Trends in cognitive sciences* 11.10 (2007) (cited on page 177).

[262] Hinton, G. E., Osindero, S., Teh, Y.-W., "A Fast Learning Algorithm for Deep Belief Nets". In: *Neural computation* 18.7 (2006) (cited on pages 28, 30, 94).

[263] Hinton, G. E., Salakhutdinov, R. R., "Reducing the Dimensionality of Data with Neural Networks". In: *Science (New York, N.Y.)* 313.5786 (2006) (cited on pages 28, 177, 179).

[264] Hinton, G. E., Zemel, R., "Autoencoders, Minimum Description Length and Helmholtz Free Energy". In: *Advances in Neural Information Processing Systems*. Volume 6. Morgan-Kaufmann, 1994 (cited on pages 177, 184).

[265] Hinton, G. E. "A Practical Guide to Training Restricted Boltzmann Machines". In: *Neural Networks: Tricks of the Trade*. Volume 7700. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. ISBN: 978-3-642-35288-1 978-3-642-35289-8 (cited on page 94).

[266] Hinton, G. E. "Training Products of Experts by Minimizing Contrastive Divergence". In: *Neural Computation* 14.8 (2002). ISSN: 0899-7667 (cited on page 179).

[267] Ho, J., Chen, X., Srinivas, A., Duan, Y., Abbeel, P., "Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design". In: *Proceedings of the 36th International Conference on Machine Learning (ICML)*. Long Beach, CA, USA, 2019 (cited on pages 46, 105).

[268]   Ho, J., Jain, A., Abbeel, P., *Denoising Diffusion Probabilistic Models*. 2020. arXiv: 2006.11239 (cited on pages 28, 30).

[269]   Hochreiter, S., Schmidhuber, J., "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997) (cited on pages 98, 106, 150).

[270]   Holzenberger, N., Du, M., Karadayi, J., Riad, R., Dupoux, E., "Learning Word Embeddings: Unsupervised Methods for Fixed-Size Representations of Variable-Length Speech Segments". In: *Annual Conference of the International Speech Communication Association*. 2018 (cited on pages 88, 200).

[271]   Hori, T., Astudillo, R., Hayashi, T., Zhang, Y., Watanabe, S., Le Roux, J., "Cycle-Consistency Training for End-to-End Speech Recognition". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019 (cited on pages 220, 221).

[272]   Hotelling, H. "Relations between Two Sets of Variates". In: *Biometrika* 28.3/4 (1936) (cited on page 198).

[273]   Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S., "Parameter-Efficient Transfer Learning for NLP". In: *International Conference on Machine Learning (ICML)*. Volume 97. 2019 (cited on page 224).

[274]   Hozjan, V., Kacic, Z., Moreno, A., Bonafonte, A., Nogueiras, A., "Interface Databases: Design and Collection of a Multilingual Emotional Speech Database". In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*. 2002 (cited on page 203).

[275]   Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., Mohamed, A., "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units". In: (2021) (cited on pages 83, 85, 87, 95, 190, 194, 207, 208).

[276]   Hsu, W.-N., Sriram, A., baevskiunsupervised$_2$021, A., Likhomanenko, T., Xu, Q., Pratap, V., Kahn, J., Lee, A., Collobert, R., Synnaeve, G., Auli, M., "Robust Wav2vec 2.0: Analyzing Domain Shift in Self-Supervised Pre-Training". In: *Proceedings of the 22nd Annual Conference of the International Speech Communication Association (Interspeech)*. ISCA, 2021 (cited on pages 216, 226).

[277]   Hsu, W.-N., Zhang, Y., Glass, J., *Learning Latent Representations for Speech Generation and Transformation*. 2017. arXiv: 1704.04222 (cited on pages 85, 92, 93, 95, 185).

[278]   Hsu, W.-N., Zhang, Y., Glass, J., "Unsupervised Learning of Disentangled and Interpretable Representations from Sequential Data". In: *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS)*. Long Beach, CA, USA, 2017 (cited on pages 38, 85, 92, 93, 95, 98, 99, 105, 159, 160, 171, 185).

[279]   Hsu, W.-N., Zhang, Y., Weiss, R. J., Zen, H., Wu, Y., Wang, Y., Cao, Y., Jia, Y., Chen, Z., Shen, J., Nguyen, P., Pang, R., "Hierarchical Generative Modeling for Controllable Speech Synthesis". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. New Orleans, LA, USA, 2019 (cited on page 38).

[280]   Hsu, Y.-C., Shen, Y., Jin, H., Kira, Z., "Generalized ODIN: Detecting out-of-Distribution Image without Learning from out-of-Distribution Data". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020 (cited on page 26).

[281]   Huang, C.-W., Tsai, S.-C., Chen, Y.-N., "PLM-ICD: Automatic ICD Coding with Pretrained Language Models". In: *Proceedings of the 4th Clinical Natural Language Processing Workshop*. Seattle, WA: Association for Computational Linguistics, 2022 (cited on pages 116, 119–122, 124, 137).

[282]   Huang, K. P., Fu, Y.-K., Zhang, Y., Lee, H.-y., "Improving Distortion Robustness of Self-Supervised Speech Processing Tasks with Domain Adaptation". In: *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*. 2022 (cited on page 226).

[283] Huang, R., Geng, A., Li, Y., "On the Importance of Gradients for Detecting Distributional Shifts in the Wild". In: *Advances in Neural Information Processing Systems* 34 (2021) (cited on page 25).

[284] Huang, R., Li, Y., "MoS: Towards Scaling Out-of-Distribution Detection for Large Semantic Space". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021 (cited on page 25).

[285] Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., Heck, L., "Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data". In: *International Conference on Information and Knowledge Management*. 2013 (cited on page 198).

[286] Huang, Y., He, L., Wei, W., Gale, W., Li, J., Gong, Y., "Using Personalized Speech Synthesis and Neural Language Generator for Rapid Speaker Adaptation". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020 (cited on page 221).

[287] Hubbard, D. W. *How to Measure Anything - Finding the Value of Intangibles in Business*. 1st edition. John Wiley & Sons Inc, 2014. 432 pages. ISBN: 978-1-118-53927-9 (cited on page 19).

[288] Huszár, F. *Is Maximum Likelihood Useful for Representation Learning?* inFERENCe. 2017. URL: https://www.inference.vc/maximum-likelihood-for-representation-learning-2/ (visited on 16 December 2021) (cited on pages 95, 109, 158, 159).

[289] Hvingelby, R., Pauli, A. B., Barrett, M., Rosted, C., Lidegaard, L. M., Søgaard, A., "DaNE: A Named Entity Resource for Danish". In: *Proceedings of the 12th Language Resources and Evaluation Conference*. 2020 (cited on page 147).

[290] "ICD-10-CM Official Guidelines for Coding and Reporting FY 2023 – UPDATED April 1, 2023 (October 1, 2022 - September 30, 2023)". In: (2023) (cited on page 10).

[291] Ilharco, G., Zhang, Y., Baldridge, J., "Large-Scale Representation Learning from Visually Grounded Untranscribed Speech". In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. 2019 (cited on page 199).

[292] Ioffe, S., Szegedy, C., "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the International Conference on Machine Learning (ICML)*. Lille, France, 2015. arXiv: 1502.03167 (cited on page 228).

[293] Ipsen, N. B., Mattei, P.-A., Frellsen, J., "Not-MIWAE: Deep Generative Modelling with Missing Not at Random Data". In: *Proceedings of the 9th International Conference on Learning Representations (ICLR)*. Virtual, 2021 (cited on pages 75, 95, 163).

[294] Jaakkola, T. S., Haussler, D., "Exploiting Generative Models in Discriminative Classifiers". In: *Advances in Neural Information Processing Systems (NIPS)* (1999) (cited on page 63).

[295] Jacobs, C., Matusevych, Y., Kamper, H., "Acoustic Word Embeddings for Zero-Resource Languages Using Self-Supervised Contrastive Learning and Multilingual Adaptation". In: *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021 (cited on page 88).

[296] Jang, E., Gu, S., Poole, B., *Categorical Reparameterization with Gumbel-Softmax*. 2016. arXiv: 1611.01144 (cited on pages 84, 186, 192).

[297] Jansen, A., Dupoux, E., Goldwater, S., Johnson, M., Khudanpur, S., Church, K., Feldman, N., Hermansky, H., Metze, F., Rose, R., "A Summary of the 2012 JHU CLSP Workshop on Zero Resource Speech Technologies and Models of Early Language Acquisition". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2013 (cited on page 222).

[298] Jansen, A., Plakal, M., Pandya, R., Ellis, D. P. W., Hershey, S., Liu, J., Moore, R. C., Saurous, R. A., "Unsupervised Learning of Semantic Audio Representations". In: *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018 (cited on page 88).

[299] Jansen, A., Thomas, S., Hermansky, H., "Weak Top-down Constraints for Unsupervised Acoustic Model Training". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vancouver, BC, Canada, 2013 (cited on page 94).

[300] Jati, A., Georgiou, P., "Neural Predictive Coding Using Convolutional Neural Networks Toward Unsupervised Learning of Speaker Characteristics". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.10 (2019). ISSN: 2329-9304 (cited on pages 82, 88, 95).

[301] Jati, A., Georgiou, P., "Speaker2Vec: Unsupervised Learning and Adaptation of a Speaker Manifold Using Deep Neural Networks with an Evaluation on Speaker Segmentation". In: *Proceedings of the 18th Annual Conference of the International Speech Communication Association (Interspeech)*. ISCA, 2017 (cited on pages 82, 88).

[302] Jayashankar, T., Roux, J. L., Moulin, P., "Detecting Audio Attacks on ASR Systems with Dropout Uncertainty". 2020. arXiv: 2006.01906 (cited on page 11).

[303] Jaynes, E. T. "Information Theory and Statistical Mechanics". In: *Physical review* 106.4 (1957) (cited on page 21).

[304] Jaynes, E. T. "Prior Probabilities". In: *IEEE Transactions on Systems Science and Cybernetics* 4.3 (1968) (cited on page 21).

[305] Jégou, H., Douze, M., Schmid, C., "Product Quantization for Nearest Neighbor Search". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.1 (2011) (cited on page 192).

[306] Jelinek, F. *Statistical Methods for Speech Recognition*. MIT Press, 1997 (cited on page 221).

[307] Ji, S., Höltţä, M., Marttinen, P., "Does the Magic of BERT Apply to Medical Code Assignment? A Quantitative Study". In: *Computers in Biology and Medicine* 139 (2021). ISSN: 0010-4825 (cited on pages 116, 137, 171).

[308] Ji, S., Sun, W., Dong, H., Wu, H., Marttinen, P., *A Unified Review of Deep Learning for Automated Medical Coding*. arXiv, 2022. arXiv: 2201.02797 (cited on page 116).

[309] Jiang, D., Lei, X., Li, W., Luo, N., Hu, Y., Zou, W., Li, X., "Improving Transformer-based Speech Recognition Using Unsupervised Pre-training". 2019. arXiv: 1910.09932 (cited on pages 83, 187, 201, 202, 207, 208).

[310] Jiang, D., Li, W., Cao, M., Zou, W., Li, X., "Speech SIMCLR: Combining Contrastive and Reconstruction Objective for Self-Supervised Speech Representation Learning". In: *Annual Conference of the International Speech Communication Association*. 2021 (cited on pages 161, 190).

[311] Jiang, D., Li, W., Zhang, R., Cao, M., Luo, N., Han, Y., Zou, W., Han, K., Li, X., "A Further Study of Unsupervised Pretraining for Transformer Based Speech Recognition". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021 (cited on pages 81, 188, 201, 202, 205, 207, 208).

[312] Jing, L., Tian, Y., "Self-Supervised Visual Feature Learning with Deep Neural Networks: A Survey". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 43.11 (2021). ISSN: 1939-3539 (cited on page 178).

[313] Johnsen, S. P., Ingeman, A., Hundborg, H. H., Schaarup, S. Z., Gyllenborg, J., "The Danish Stroke Registry". In: *Clinical Epidemiology* (2016) (cited on pages 140, 148).

[314] Johnson, A. E. W., Bulgarelli, L., Shen, L., Gayles, A., Shammout, A., Horng, S., Pollard, T. J., Moody, B., Gow, B., Lehman, L.-w. H., Celi, L. A., Mark, R. G., "MIMIC-IV, a Freely Accessible Electronic Health Record Dataset". In: *Scientific Data* 10.1 (2023). ISSN: 2052-4463 (cited on pages 10, 116, 117, 138, 171).

[315] Johnson, A. E. W., Pollard, T. J., Shen, L., Lehman, L.-w. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., Mark, R. G., "MIMIC-III, a Freely Accessible Critical Care Database". In: *Scientific Data* 3.1 (2016). ISSN: 2052-4463 (cited on pages 10, 117, 171).

[316] Jones, H. T., Moore, J., "Is the Discrete VAE's Power Stuck in Its Prior?" In: *"I Can't Believe It's Not Better!" NeurIPS 2020 Workshop*. 2020 (cited on page 94).

[317] Jordan, M. I., Jacobs, R. A., "Hierarchical Mixtures of Experts and the EM Algorithm". In: *Neural Computation* 6.2 (1994) (cited on page 177).

[318] Jordan, M. I., Mitchell, T. M., "Machine Learning: Trends, Perspectives, and Prospects". In: *Science (New York, N.Y.)* 349.6245 (2015) (cited on page 177).

[319] Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., Saul, L. K., "An Introduction to Variational Methods for Graphical Models". In: *Machine Learning* 37.2 (1999) (cited on pages 31, 90).

[320] Kahn, J., Riviere, M., Zheng, W., Kharitonov, E., Xu, Q., Mazare, P., Karadayi, J., Liptchinsky, V., Collobert, R., Fuegen, C., al. e., "Libri-Light: A Benchmark for ASR with Limited or No Supervision". In: *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Virtual, 2020 (cited on pages 105, 200, 242).

[321] Kamper, H. "Truly Unsupervised Acoustic Word Embeddings Using Weak Top-down Constraints in Encoder-Decoder Models". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019 (cited on pages 88, 200).

[322] Kamper, H., Elsner, M., Jansen, A., Goldwater, S., "Unsupervised Neural Network Based Feature Extraction Using Weak Top-down Constraints". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015 (cited on pages 88, 186, 200).

[323] Kamper, H., Jansen, A., Goldwater, S., "A Segmental Framework for Fully-Unsupervised Large-Vocabulary Speech Recognition". In: *Computer Speech & Language* 46 (2017). ISSN: 0885-2308 (cited on pages 200, 218).

[324] Kamper, H., Livescu, K., Goldwater, S., "An Embedded Segmental K-means Model for Unsupervised Segmentation and Clustering of Speech". In: *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. 2017 (cited on page 200).

[325] Kamper, H., Roth, M., "Visually Grounded Cross-Lingual Keyword Spotting in Speech". In: *Proceedings of the 6th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU)*. 2018 (cited on page 199).

[326] Kamper, H., Niekerk, B., "Towards Unsupervised Phone and Word Segmentation Using Self-Supervised Vector-Quantized Neural Networks". In: *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)* (2021) (cited on pages 87, 89).

[327] Kamper, H., Wang, W., Livescu, K., "Deep Convolutional Acoustic Word Embeddings Using Word-Pair Side Information". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016 (cited on page 200).

[328] Karras, T., Laine, S., Aila, T., "A Style-Based Generator Architecture for Generative Adversarial Networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 (cited on page 30).

[329] Katan, M., Luft, A., "Global Burden of Stroke". In: *Seminars in Neurology*. Volume 38. 02. Thieme Medical Publishers, 2018 (cited on pages 8, 139).

[330] Kavuluru, R., Rios, A., Lu, Y., "An Empirical Evaluation of Supervised Learning Approaches in Assigning Diagnosis Codes to Electronic Medical Records". In: *Artificial Intelligence in Medicine*. Intelligent Healthcare Informatics in Big Data Era 65.2 (2015). ISSN: 0933-3657 (cited on pages 116, 124).

[331] Kawakami, K., Wang, L., Dyer, C., Blunsom, P., Oord, A., "Learning Robust and Multilingual Speech Representations". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2020). arXiv: 2001.11128 (cited on pages 83, 190, 201, 202, 204–206, 208, 215, 216).

[332] Kemp, T., Waibel, A., "Unsupervised Training of a Speech Recognizer: Recent Experiments". In: *In Proceedings of the European Conference on Speech Communication and Technology (EUROSPEECH)*. 1999 (cited on pages 176, 181).

[333] Kendall, A., Gal, Y., "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In: *Advances in Neural Information Processing Systems*. Volume 30. Curran Associates, Inc., 2017 (cited on page 12).

[334] Kharitonov, E., Lee, A., Polyak, A., Adi, Y., Copet, J., Lakhotia, K., Nguyen, T. A., Rivière, M., Mohamed, A., Dupoux, E., Hsu, W.-N., *Text-Free Prosody-Aware Generative Spoken Language Modeling*. 2021. arXiv: 2109.03264 (cited on pages 185, 223, 226).

[335] Khurana, S., Joty, S. R., Ali, A., Glass, J., "A Factorial Deep Markov Model for Unsupervised Disentangled Representation Learning from Speech". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brighton, United Kingdom: IEEE, 2019. ISBN: 978-1-4799-8131-1 (cited on pages 85, 92, 93, 95).

[336] Khurana, S., Laurent, A., Glass, J., "Magic Dust for Cross-Lingual Adaptation of Monolingual Wav2vec-2.0". 2021. arXiv: 2110.03560 (cited on page 89).

[337] Khurana, S., Laurent, A., Hsu, W.-N., Chorowski, J., Lancucki, A., Marxer, R., Glass, J., *A Convolutional Deep Markov Model for Unsupervised Speech Representation Learning*. 2020. arXiv: 2006.02547 (cited on pages 85, 92–94, 207).

[338] Kim, B.-H., Ganapathi, V., "Read, Attend, and Code: Pushing the Limits of Medical Codes Prediction from Clinical Notes by Machines". In: *Proceedings of the 6th Machine Learning for Healthcare Conference*. PMLR, 2021 (cited on pages 116, 120, 121, 136, 171).

[339] Kingma, D. P., Dhariwal, P., "Glow: Generative Flow with Invertible 1×1 Convolutions". In: *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS)*. Montréal, Canada, 2018 (cited on pages 30, 42, 47, 52, 69, 263).

[340] Kingma, D. P., Welling, M., "Auto-Encoding Variational Bayes". In: *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*. Banff, AB, Canada, 2014. arXiv: 1312.6114 (cited on pages 28, 30, 31, 35, 38, 42, 45, 61, 69, 89, 90, 97, 100, 180, 185).

[341] Kingma, D. P., Ba, J., *Adam: A Method for Stochastic Optimization*. arXiv, 2017. arXiv: 1412.6980 (cited on page 124).

[342] Kingma, D. P., Ba, J. L., "Adam: A Method for Stochastic Optimization". In: *Proceedings of the the 3rd International Conference for Learning Representations (ICLR)*. San Diego, CA, USA, 2015. arXiv: 1412.6980 (cited on pages 230, 244, 245, 259).

[343] Kingma, D. P., Rezende, D. J., Mohamed, S., Welling, M., "Semi-Supervised Learning with Deep Generative Models". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems (NeurIPS)*. Montréal, Quebec, Canada, 2014. arXiv: 1406.5298 (cited on pages 38, 57, 98, 161).

[344] Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., Welling, M., "Improved Variational Inference with Inverse Autoregressive Flow". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems (NeurIPS)*. NIPS'16. Barcelona, Spain, 2016. ISBN: 978-1-5108-3881-9 (cited on pages 38, 45, 51, 92, 161).

[345] Kingma, D. P., Salimans, T., Poole, B., Ho, J., *Variational Diffusion Models*. 2021. arXiv: 2107.00630 (cited on page 97).

[346] Kipf, T. N., Welling, M., *Variational Graph Auto-Encoders*. 2016. arXiv: 1611.07308 (cited on page 42).

[347] Kirichenko, P., Izmailov, P., Wilson, A. G., *Why Normalizing Flows Fail to Detect Out-of-Distribution Data*. 2020. arXiv: 2006.08545 (cited on pages 24, 28, 68, 69, 163, 263, 264, 271–273).

[348] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., Hadsell, R., "Overcoming Catastrophic Forgetting in Neural Networks". In: *Proceedings of the National Academy of Sciences (PNAS)* (2017) (cited on page 259).

[349] Klejch, O., Wallington, E., Bell, P., "Deciphering Speech: A Zero-Resource Approach to Cross-Lingual Transfer in Asr". In: *Proceedings of the 23rd Annual Conference of the International Speech Communication Association (Interspeech)*. Incheon, South Korea: ISCA, 2022 (cited on pages 218, 219).

[350] Kocabiyikoglu, A. C., Besacier, L., Kraif, O., "Augmenting LibriSpeech with French Translations: A Multimodal Corpus for Direct Speech Translation Evaluation". In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*. 2018 (cited on page 203).

[351] Koenecke, A., Nam, A., Lake, E., Nudell, J., Quartey, M., Mengesha, Z., Toups, C., Rickford, J. R., Jurafsky, D., Goel, S., "Racial Disparities in Automated Speech Recognition". In: *Proceedings of the National Academy of Sciences* 117.14 (2020) (cited on page 242).

[352] Kramer, M. A. "Nonlinear Principal Component Analysis Using Autoassociative Neural Networks". In: *AIChE journal* 37.2 (1991) (cited on page 88).

[353] Krebes, S., Ebinger, M., Baumann, A. M., Kellner, P. A., Rozanski, M., Doepp, F., Sobesky, J., Gensecke, T., Leidel, B. A., Malzahn, U., "Development and Validation of a Dispatcher Identification Algorithm for Stroke Emergencies". In: *Stroke* 43.3 (2012) (cited on page 140).

[354] Kreuk, F., Keshet, J., Adi, Y., *Self-Supervised Contrastive Learning for Unsupervised Phoneme Segmentation*. 2020. arXiv: 2007.13465. (Visited on 23 September 2023) (cited on page 89).

[355] Kreuk, F., Polyak, A., Copet, J., Kharitonov, E., Nguyen, T. A., Rivière, M., Hsu, W.-N., Mohamed, A., Dupoux, E., Adi, Y., "Textless Speech Emotion Conversion Using Discrete & Decomposed Representations". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, 2022 (cited on page 223).

[356] Krizhevsky, A. "Learning Multiple Layers of Features from Tiny Images". University of Toronto, 2009. 1-60. ISBN: 9788578110796. pmid: 25246403 (cited on pages 50, 69, 105, 227).

[357] Krizhevsky, A., Sutskever, I., Hinton, G. E., "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings of the Conference on Neural Information Processing Systems (NIPS)* (2012). ISSN: 10495258. pmid: 7491034 (cited on page 157).

[358] Kull, M., Perello-Nieto, M., Kängsepp, M., Filho, T. S., Song, H., Flach, P., "Beyond Temperature Scaling: Obtaining Well-Calibrated Multiclass Probabilities with Dirichlet Calibration". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS)*. Red Hook, NY, USA: Curran Associates Inc., 2019 (cited on page 11).

[359] Kullback, S. *Information Theory and Statistics*. John Wiley & Sons, 1959. ISBN: 0-8446-5625-9 (cited on page 21).

[360] Kyu, H. H., Abate, D., Abate, K. H., Abay, S. M., Abbafati, C., Abbasi, N., Abbastabar, H., Abd-Allah, F., Abdela, J., Abdelalim, A., "Global, Regional, and National Disability-Adjusted Life-Years (DALYs) for 359 Diseases and Injuries and Healthy Life Expectancy (HALE) for 195 Countries and Territories, 1990–2017: A Systematic Analysis for the Global Burden of Disease Study 2017". In: *The Lancet* 392.10159 (2018) (cited on pages 8, 139).

[361] Lai, C.-I., Chuang, Y.-S., Lee, H.-Y., Li, S.-W., Glass, J., "Semi-Supervised Spoken Language Understanding via Self-Supervised Speech and Language Model Pretraining". In: *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021 (cited on page 95).

[362] Lai, C.-I. J., Zhang, Y., Liu, A. H., Chang, S., Liao, Y.-L., Chuang, Y.-S., Qian, K., Khurana, S., Cox, D., Glass, J., "PARP: Prune, Adjust and Re-Prune for Self-Supervised Speech Recognition". In: *Conference on Neural Information Processing Systems (Neurips)*. 2021 (cited on page 225).

[363] Lai, G., Dai, Z., Yang, Y., Yoo, S., "Re-Examination of the Role of Latent Variables in Sequence Modeling". In: *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS)*. Vancouver, BC, Canada, 2019 (cited on page 109).

[364] Lai, G., Li, B., Zheng, G., Yang, Y., *Stochastic WaveNet: A Generative Latent Variable Model for Sequential Data*. 2018. arXiv: 1806.06116 (cited on pages 99, 105, 246, 249).

[365] Lai, P. L., Fyfe, C., "A Neural Implementation of Canonical Correlation Analysis". In: *Neural Networks* 12.10 (1999) (cited on page 198).

[366] Lai, P. L., Fyfe, C., "Kernel and Nonlinear Canonical Correlation Analysis". In: *International Journal of Neural Systems* 10.5 (2000) (cited on page 198).

[367] Lake, B. M., Salakhutdinov, R., Tenenbaum, J. B., "Human-Level Concept Learning through Probabilistic Program Induction". In: *Science* 350.6266 (2015). ISSN: 0036-8075 (cited on page 227).

[368] Lakshminarayanan, B., Pritzel, A., Blundell, C., "Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles". In: *In Proceddings of the 31st Conference on Neural Information Processing Systems (NeurIPS)*. Long Beach, CA, USA, 2017 (cited on pages 13, 25, 41, 47, 53).

[369] Laleye, F. A. A., Besacier, L., Ezin, E. C., Motamed, C., "First Automatic Fongbe Continuous Speech Recognition System: Development of Acoustic Models and Language Models". In: *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems (FedC-SIS)*. Volume 8. Annals of Computer Science and Information Systems. Gdańsk, Poland: IEEE, 2016 (cited on page 204).

[370] Lamel, L., Gauvain, J.-L., Adda, G., "Lightly Supervised and Unsupervised Acoustic Model Training". In: *Computer Speech & Language* 16.1 (2002) (cited on pages 176, 181).

[371] Lample, G., Conneau, A., Denoyer, L., Ranzato, M., "Unsupervised Machine Translation Using Monolingual Corpora Only". In: *International Conference on Learning Representations (ICLR)* (2018) (cited on page 219).

[372] Laptev, A., Korostik, R., Svischev, A., Andrusenko, A., Medennikov, I., Rybin, S., "You Do Not Need More Data: Improving End-to-End Speech Recognition by Text-to-Speech Data Augmentation". In: *Proceedings of the International Congress on Image and Signal Processing, Biomedical Engineering and Informatics (CISP-BMEI)*. 2020 (cited on page 221).

[373] Latif, S., Rana, R., Khalifa, S., Jurdak, R., Qadir, J., Schuller, B. W., *Deep Representation Learning in Speech Processing: Challenges, Recent Advances, and Future Trends*. 2021. arXiv: 2001.00378 (cited on page 178).

[374] LeCun, Y., Huang, F., Bottou, L., "Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Volume 2. 2004 (cited on page 227).

[375] LeCun, Y. "The MNIST Database of Handwritten Digits". In: *http://yann. lecun. com/exdb/mnist/* (1998) (cited on page 69).

[376] LeCun, Y., Bengio, Y., "Convolutional Networks for Images, Speech, and Time Series". In: *The handbook of brain theory and neural networks* 3361.10 (1995) (cited on page 150).

[377] LeCun, Y., Bengio, Y., Hinton, G., "Deep Learning". In: *Nature Publishing Group, UK, London* 521.7553 (2015) (cited on pages 5, 36, 176).

[378] LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F., "A Tutorial on Energy-Based Learning". In: *Predicting structured data* 1.0 (2006) (cited on pages 26, 30, 177).

[379] Lecun, Y., Soulie Fogelman, F., "Modeles Connexionnistes de l'apprentissage". In: *Intellectica, special issue apprentissage et machine* 2 (1987) (cited on page 259).

[380] LeCun, Y. A., Bottou, L., Bengio, Y., Haffner, P., "Gradient-Based Learning Applied to Document Recognition". In: *Proceedings of the IEEE* 86.11 (1998) (cited on pages 50, 105, 227).

[381] Lee, A., Gong, H., Duquenne, P.-A., Schwenk, H., Chen, P.-J., Wang, C., Popuri, S., Adi, Y., Pino, J. M., Gu, J., Hsu, W.-N., "Textless Speech-to-Speech Translation on Real Data". In: *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*. Seattle, WA, USA: Association for Computational Linguistics, 2022 (cited on page 224).

[382] Lee, B., Hasegawa-Johnson, M., Goudeseune, C., Kamdar, S., Borys, S., Liu, M., Huang, T., "AVICAR: Audio-visual Speech Corpus in a Car Environment". In: *Eighth International Conference on Spoken Language Processing*. 2004 (cited on page 198).

[383] Lee, C.-y., Glass, J., "A Nonparametric Bayesian Approach to Acoustic Model Discovery". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Jeju Island, Korea: Association for Computational Linguistics, 2012 (cited on page 94).

[384] Lee, D. D., Seung, H. S., "Learning the Parts of Objects by Non-Negative Matrix Factorization". In: *Nature* 401.6755 (1999) (cited on page 177).

[385] Lee, H., Battle, A., Raina, R., Ng, A. Y., "Efficient Sparse Coding Algorithms". In: *Proceedings of the Conference on Neural Information Processing Systems (NIPS)* 19.2 (2006). ISSN: 10495258. pmid: 17051527 (cited on page 179).

[386] Lee, H., Largman, Y., Pham, P., Ng, A. Y., "Unsupervised Feature Learning for Audio Classification Using Convolutional Deep Belief Networks". In: *Proceedings of the 22nd International Conference on Neural Information Processing Systems*. NIPS'09. Vancouver, BC, Canada, 2009. ISBN: 978-1-61567-911-9 (cited on pages 94, 95).

[387] Lee, J., Scott, D. J., Villarroel, M., Clifford, G. D., Saeed, M., Mark, R. G., "Open-Access MIMIC-II Database for Intensive Care Research". In: *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference* 2011 (2011). ISSN: 1557-170X. pmid: 22256274 (cited on page 117).

[388] Lee, K., Lee, H., Lee, K., Shin, J., "Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples". In: *Proceedings of the 6th International Conference on Learning Representations (ICLR)*. 2018 (cited on page 25).

[389] Lee, K., Lee, K., Lee, H., Shin, J., "A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks". In: *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS)*. Montréal, Quebec, Canada, 2018 (cited on pages 24, 25, 27, 47, 53, 63, 163, 260).

[390] Legerstee, M. "Infants Use Multimodal Information to Imitate Speech Sounds". In: *Infant Behavior and Development* 13.3 (1990) (cited on page 197).

[391] Lemonte, A. *The Gradient Test: Another Likelihood-Based Test*. Academic Press, 2016 (cited on page 61).

[392] Levin, K., Henry, K., Jansen, A., Livescu, K., "Fixed-Dimensional Acoustic Embeddings of Variable-Length Segments in Low-Resource Settings". In: *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. 2013 (cited on pages 88, 200).

[393]   Levin, K., Jansen, A., Van Durme, B., "Segmental Acoustic Indexing for Zero Resource Keyword Search". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015 (cited on page 199).

[394]   Lewis, D. D. "A Sequential Algorithm for Training Text Classifiers: Corrigendum and Additional Data". In: *Acm Sigir Forum*. Volume 29. 2. ACM New York, NY, USA, 1995 (cited on page 11).

[395]   Lhoest, Q. *Datasets: A Community Library for Natural Language Processing*. 2021. arXiv: 2109.02846. (Visited on 31 December 2023) (cited on page 164).

[396]   Li, D., Chen, D., Goh, J., Ng, S.-k., *Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series*. 2019. arXiv: 1809.04758. (Visited on 17 September 2023) (cited on page 24).

[397]   Li, D., Chen, D., Jin, B., Shi, L., Goh, J., Ng, S.-K., "MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks". In: *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*. Volume 11730. Munich, Germany: Springer, 2019 (cited on pages 24, 29).

[398]   Li, F., Yu, H., "ICD Coding from Clinical Text Using Multi-Filter Residual Convolutional Neural Network". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.05 (2020). ISSN: 2374-3468 (cited on pages 116, 119–122, 136, 137).

[399]   Li, J., Gadde, R., Ginsburg, B., Lavrukhin, V., *Training Neural Speech Recognition Systems with Synthetic Speech Augmentation*. 2018. arXiv: 1811.00707. (Visited on 12 April 2023) (cited on page 221).

[400]   Liang, S., Li, Y., Srikant, R., "Enhancing the Reliability of Out-of-Distribution Image Detection in Neural Networks". In: *Proceedings of the 6th International Conference on Learning Representations (ICLR)*. Vancouver, Canada, 2018 (cited on pages 25, 47, 53, 56).

[401]   Likhomanenko, T., Xu, Q., Kahn, J., Synnaeve, G., Collobert, R., "slimIPL: Language-model-free Iterative Pseudo-Labeling". In: *Annual Conference of the International Speech Communication Association*. 2021 (cited on pages 209, 210).

[402]   Lin, G.-T., Chuang, Y.-S., Chung, H.-L., Yang, S.-W., Chen, H.-J., Dong, S. A., Li, S.-W., Mohamed, A., Lee, H.-y., Lee, L.-S., "DUAL: Discrete Spoken Unit Adaptive Learning for Textless Spoken Question Answering". In: *IEEE Annual Conference of the International Speech Communication Association (Interspeech)*. Incheon, South Korea: ISCA, 2022 (cited on page 224).

[403]   Lin, G.-T., Hsu, C.-J., Liu, D.-R., Lee, H.-Y., Tsao, Y., "Analyzing the Robustness of Unsupervised Speech Recognition". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Singapore: IEEE, 2022 (cited on page 220).

[404]   Lin, J., Nogueira, R., Yates, A., "Pretrained Transformers for Text Ranking: BERT and Beyond". 2021. arXiv: 2010.06467 (cited on pages 137, 171).

[405]   Ling, S., Liu, Y., "DeCoAR 2.0: Deep Contextualized Acoustic Representations with Vector Quantization". 2020. arXiv: 2012.06659 (cited on pages 83, 85, 87, 95, 188, 190, 207).

[406]   Ling, S., Liu, Y., Salazar, J., Kirchhoff, K., "Deep Contextualized Acoustic Representations for Semi-Supervised Speech Recognition". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020 (cited on pages 83, 186, 190, 206).

[407]   Ling, S., Salazar, J., Liu, Y., Kirchhoff, K., "BERTphone: Phonetically-aware Encoder Representations for Utterance-Level Speaker and Language Recognition". In: 2020 (cited on page 213).

[408]   Liu, A. H., Chung, Y.-A., Glass, J., "Non-Autoregressive Predictive Coding for Learning Speech Representations from Local Dependencies". 2020. arXiv: 2011.00406 (cited on pages 84, 85, 95, 188, 190, 207).

[409] Liu, A. H., Hsu, W.-N., Auli, M., Baevski, A., "Towards End-to-End Unsupervised Speech Recognition". In: *IEEE Spoken Language Technology Workshop (SLT)*. Doha, Qatar: IEEE, 2023 (cited on pages 218–220).

[410] Liu, A. T., Li, S.-W., Lee, H.-y., "TERA: Self-Supervised Learning of Transformer Encoder Representation for Speech". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2021). ISSN: 2329-9304 (cited on pages 82, 95, 188, 190, 207).

[411] Liu, A. T., Yang, S.-w., Chi, P.-H., Hsu, P.-c., Lee, H.-y., "Mockingjay: Unsupervised Speech Representation Learning with Deep Bidirectional Transformer Encoders". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020 (cited on pages 83, 85, 95, 187, 188, 190, 206, 215).

[412] Liu, F., Xu, W., Lu, J., Zhang, G., Gretton, A., Sutherland, D. J., "Learning Deep Kernels for Non-Parametric Two-Sample Tests". In: *International Conference on Machine Learning (ICML)*. PMLR, 2020 (cited on page 63).

[413] Liu, H., Son, K., Yang, J., Liu, C., Gao, J., Lee, Y. J., Li, C., "Learning Customized Visual Models with Retrieval-Augmented Knowledge". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Vancouver, BC, Canada: IEEE, 2023 (cited on page 161).

[414] Liu, L., Huang, Y., *Masked Pre-Trained Encoder Base on Joint Ctc-Transformer*. 2020. arXiv: 2005.11978 (cited on pages 187, 190, 207).

[415] Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G., *Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing*. 2021. arXiv: 2107.13586 (cited on pages 178, 224).

[416] Liu, D.-R., Chen, K.-Y., Lee, H.-Y., Lee, L.-s., "Completely Unsupervised Phoneme Recognition by Adversarially Learning Mapping Relationships from Audio Embeddings". In: *Proceedings of the 19th Annual Conference of the International Speech Communication Association (Interspeech)*. ISCA, 2018 (cited on pages 218, 219).

[417] Liu, W., Wang, X., Owens, J. D., Li, Y., *Energy-Based Out-of-distribution Detection*. 2020. arXiv: 2010.03759 (cited on pages 23, 24, 26).

[418] Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., Tang, J., "Self-Supervised Learning: Generative or Contrastive". In: *IEEE Transactions on Knowledge & Data Engineering* 01 (2021). ISSN: 1558-2191 (cited on page 178).

[419] Liu, Y., Cheng, H., Klopfer, R., Gormley, M. R., Schaaf, T., "Effective Convolutional Attention Network for Multi-label Clinical Document Classification". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021 (cited on pages 116, 119, 137).

[420] Liu, Y., Fung, P., Yang, Y., Cieri, C., Huang, S., Graff, D., "HKUST/MTS: A Very Large Scale Mandarin Telephone Speech Corpus". In: *Proceedings of the International Conference on Spoken Language Processing*. 2006 (cited on page 202).

[421] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 (cited on page 180).

[422] Liu, Y., Chen, J., Deng, L., "Unsupervised Sequence Classification Using Sequential Output Statistics". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2017 (cited on page 218).

[423] Liu, Z., Zhou, J. P., Wang, Y., Weinberger, K. Q., "Unsupervised Out-of-Distribution Detection with Diffusion Inpainting". In: *Proceedings of the International Conference on Machine Learning (ICML)*. Volume 202. Proceedings of Machine Learning Research. Honolulu, Hawaii, USA: PMLR, 2023. arXiv: 2302.10326 (cited on pages 23, 24, 29).

[424] Liu, Z., Luo, P., Wang, X., Tang, X., "Deep Learning Face Attributes in the Wild". In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2015 (cited on pages 71, 105).

[425] Lloyd, S. S., Rissing, J. P., "Physician and Coding Errors in Patient Records". In: *JAMA : the journal of the American Medical Association* 254.10 (1985). issn: 0098-7484 (cited on page 138).

[426] Loshchilov, I., Hutter, F., "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2022 (cited on page 124).

[427] Lugosch, L., Ravanelli, M., Ignoto, P., Tomar, V. S., Bengio, Y., "Speech Model Pre-Training for End-to-End Spoken Language Understanding". In: *Annual Conference of the International Speech Communication Association*. 2019 (cited on pages 203, 211).

[428] Lundervold, A. S., Lundervold, A., "An Overview of Deep Learning in Medical Imaging Focusing on MRI". In: *Zeitschrift für Medizinische Physik* 29.2 (2019) (cited on page 5).

[429] Luo, J., Wang, J., Cheng, N., Xiao, J., "Dropout Regularization for Self-Supervised Learning of Transformer Encoder Speech Representation". In: *Annual Conference of the International Speech Communication Association* (2021) (cited on page 188).

[430] Luo, Y., Mesgarani, N., "Conv-TasNet: Surpassing Ideal Time-Frequency Magnitude Masking for Speech Separation". In: *IEEE ACM Trans. Audio Speech Lang. Process.* 27.8 (2019) (cited on page 243).

[431] Lyudchik, O. *Outlier Detection Using Autoencoders*. 2016 (cited on pages 24, 29).

[432] Ma, C., Tschiatschek, S., Palla, K., Hernández-Lobato, J. M., Nowozin, S., Zhang, C., "EDDI: Efficient Dynamic Discovery of High-Value Information with Partial VAE". In: *Proceedings of the 36th International Conference on Machine Learning, (ICML)*. PMLR, 2019 (cited on page 75).

[433] Ma, J., Matsoukas, S., Kimball, O., Schwartz, R., "Unsupervised Training on Large Amounts of Broadcast News Data". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Volume 3. 2006 (cited on pages 176, 181).

[434] Maaløe, L., Fraccaro, M., Liévin, V., Winther, O., "BIVA: A Very Deep Hierarchy of Latent Variables for Generative Modeling". In: *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS)*. Vancouver, Canada, 2019 (cited on pages 24, 28, 37, 38, 43, 44, 46–48, 50, 52, 55, 69, 97, 99, 105, 159–161, 170, 228, 229, 250).

[435] Maaløe, L., Fraccaro, M., Winther, O., *Semi-Supervised Generation with Cluster-aware Generative Models*. 2017. arXiv: 1704.00637 (cited on page 57).

[436] Maaløe, L., Sønderby, C. K., Sønderby, S. K., Winther, O., "Auxiliary Deep Generative Models". In: *Proceedings of the 33rd International Conference on Machine Learning (ICML)*. Volume 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 2016 (cited on page 57).

[437] Maas, A. L., Miller, S. D., O'neil, T. M., Ng, A. Y., Nguyen, P., "Word-Level Acoustic Modeling with Convolutional Vector Regression". In: *Procedings of the International Conference on Machine Learning (ICML), Workshop on Representation Learning*. 2012 (cited on page 200).

[438] MacKay, D. J. C. "A Practical Bayesian Framework for Backpropagation Networks". In: *Neural Computation* 4.3 (1992). issn: 0899-7667 (cited on page 13).

[439] MacKay, D. J. C. *Information Theory, Inference, and Learning Algorithms*. 1st edition. Cambridge University Press, 2003. 640 pages. isbn: 978-0-521-64298-9 (cited on pages 19, 57).

[440] Maddison, C. J., Mnih, A., Teh, Y. W., "The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables". In: *Proceedings of the 5th International Conference on Learning Representations (ICLR)*. Toulon, France, 2017 (cited on page 84).

[441] Maekaku, T., Chang, X., Fujita, Y., Chen, L.-W., Watanabe, S., Rudnicky, A., "Speech Representation Learning Combining Conformer CPC with Deep Cluster for the ZeroSpeech Challenge 2021". In: *IEEE Annual Conference of the International Speech Communication Association (Interspeech)*. 2021 (cited on page 223).

[442] Magic Data Technology Co., Ltd. *MAGICDATA Mandarin Chinese Read Speech Corpus*. 2019 (cited on page 202).

[443] Martens, J. "New Insights and Perspectives on the Natural Gradient Method". In: *Journal of Machine Learning Research* (2020) (cited on pages 259–261).

[444] Martens, J., Grosse, R., "Optimizing Neural Networks with Kronecker-Factored Approximate Curvature". In: *International Conference on Machine Learning (ICML)*. PMLR, 2015 (cited on page 66).

[445] Masumura, R., Makishima, N., Ihori, M., Takashima, A., Tanaka, T., Orihashi, S., "Phoneme-to-Grapheme Conversion Based Large-Scale Pre-Training for End-to-End Automatic Speech Recognition". In: *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*. 2020 (cited on page 221).

[446] Mattei, P.-A., Frellsen, J., "Leveraging the Exact Likelihood of Deep Latent Variable Models". In: *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS)*. Montreal, Canada, 2018 (cited on pages 105, 246).

[447] Mattei, P.-A., Frellsen, J., "MIWAE: Deep Generative Modelling and Imputation of Incomplete Data Sets". In: *Proceedings of the 36th International Conference on Machine Learning (ICML)*. Long Beach, CA, USA, 2019 (cited on pages 75, 95, 163).

[448] Mattei, P.-A., Frellsen, J., "Refit Your Encoder When New Data Comes By". In: *3rd NeurIPS Workshop on Bayesian Deep Learning*. 2018 (cited on pages 47, 62).

[449] Melzer, T., Reiter, M., Bischof, H., "Nonlinear Feature Extraction Using Generalized Canonical Correlation Analysis". In: *International Conference on Artificial Neural Networks*. 2001 (cited on page 198).

[450] Merkx, D., Frank, S. L., Ernestus, M., "Language Learning Using Speech to Image Retrieval". In: *Annual Conference of the International Speech Communication Association* (2019) (cited on page 199).

[451] Mesaros, A., Heittola, T., Virtanen, T., "Detection and Classification of Acoustic Scenes and Events". In: *Proceedings of the Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)* (2018) (cited on pages 204, 205).

[452] Metz, C. "How Safe Are Systems Like Tesla's Autopilot? No One Knows." In: *The New York Times. Technology* (2022). ISSN: 0362-4331 (cited on page 7).

[453] Michaeli, T., Wang, W., Livescu, K., "Nonparametric Canonical Correlation Analysis". In: *Proceedings of the International Conference on Machine Learning (ICML)*. 2016 (cited on page 198).

[454] Michalopoulos, G., Malyska, M., Sahar, N., Wong, A., Chen, H., "ICDBigBird: A Contextual Embedding Model for ICD Code Classification". In: *Proceedings of the 21st Workshop on Biomedical Language Processing*. Dublin, Ireland: Association for Computational Linguistics, 2022 (cited on pages 116, 137, 171).

[455] Michel, P., Rasanen, O., Thiollière, R., Dupoux, E., "Blind Phoneme Segmentation with Temporal Prediction Errors". In: *Association for Computational Linguistics (ACL) Student Research Workshop* (2017) (cited on page 89).

[456] Mikolov, T., Chen, K., Corrado, G., Dean, J., "Efficient Estimation of Word Representations in Vector Space". In: *Workshop Track Proceedings of the International Conference on Learning Representations (ICLR)*. Scottsdale, Arizona, USA, 2013. arXiv: 1301.3781 (cited on pages 28, 38).

[457]   Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., Khudanpur, S., "Recurrent Neural Network Based Language Model." In: *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*. Volume 2. 3. Makuhari, 2010 (cited on page 81).

[458]   Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J., "Distributed Representations of Words and Phrases and Their Compositionality". In: *Proceedings of the Conference on Neural Information Processing Systems (NIPS)* (2013). ISSN: 10495258. pmid: 903 (cited on pages 88, 94).

[459]   Milde, B., Biemann, C., *Unspeech: Unsupervised Speech Context Embeddings*. 2018. arXiv: 1804.0677. (Visited on 21 October 2021) (cited on pages 85, 88, 94, 95, 190).

[460]   Mnih, A., Gregor, K., "Neural Variational Inference and Learning in Belief Networks". In: *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 2014. arXiv: 1402.0030v2 (cited on page 29).

[461]   Mohamed, A., Lee, H.-y., Borgholt, L., **Havtorn, J. D.**, Edin, J., Igel, C., Kirchhoff, K., Li, S.-W., Livescu, K., Maaløe, L., Sainath, T. N., Watanabe, S., "Self-Supervised Speech Representation Learning: A Review". In: *IEEE Journal of Selected Topics in Signal Processing (JSTSP)* 16.6 (2022). arXiv: 2205.10643 (cited on pages viii, 16, 137, 147, 171, 175).

[462]   Mohamed, S., Rosca, M., Figurnov, M., Mnih, A., *Monte Carlo Gradient Estimation in Machine Learning*. 2019. arXiv: 1906.10652 (cited on pages 35, 90).

[463]   Mohri, M., Pereira, F., Riley, M., "Weighted Finite-State Transducers in Speech Recognition". In: *Computer Speech & Language* 16.1 (2002) (cited on page 221).

[464]   Moons, E., Khanna, A., Akkasi, A., Moens, M.-F., "A Comparison of Deep Learning Methods for ICD Coding of Clinical Records". In: *Applied Sciences* 10.15 (2020). ISSN: 2076-3417 (cited on pages 116, 137).

[465]   Morais, E., Kuo, H.-K. J., Thomas, S., Tüske, Z., Kingsbury, B., "End-to-End Spoken Language Understanding Using Transformer Networks and Self-Supervised Pre-Trained Features". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021 (cited on page 95).

[466]   Moreno-Muñoz, P., Recasens, P. G., Hauberg, S., *On Masked Pre-training and the Marginal Likelihood*. 2023. arXiv: 2306.00520. (Visited on 18 October 2023) (cited on page 170).

[467]   Morningstar, W., Ham, C., Gallagher, A., Lakshminarayanan, B., Alemi, A., Dillon, J., "Density of States Estimation for Out of Distribution Detection". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2021 (cited on pages 24, 29, 64, 68, 69, 72, 271–273).

[468]   Muirhead, R. J. *Aspects of Multivariate Statistical Theory*. Volume 197. John Wiley & Sons, 2009 (cited on page 231).

[469]   Mullenbach, J., Wiegreffe, S., Duke, J., Sun, J., Eisenstein, J., "Explainable Prediction of Medical Codes from Clinical Text". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, 2018 (cited on pages 117–122, 126, 136, 171).

[470]   Nagrani, A., Chung, J. S., Zisserman, A., "VoxCeleb: A Large-Scale Speaker Identification Dataset". In: *Annual Conference of the International Speech Communication Association*. 2017 (cited on page 203).

[471]   Nair, V., Hinton, G. E., "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *Proceedings of the 27th International Conference on Machine Learning (ICML 2021)*. Haifa, Israel, 2010 (cited on page 228).

[472] Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., Lakshminarayanan, B., "Do Deep Generative Models Know What They Don't Know?" In: *Proceedings of the 7th International Conference on Learning Representations (ICLR)*. New Orleans, LA, USA, 2019. arXiv: 1810.09136 (cited on pages 42, 43, 50, 52, 54, 57, 60, 63, 68, 69, 263).

[473] Nalisnick, E., Matsukawa, A., Teh, Y. W., Lakshminarayanan, B., *Detecting Out-of-Distribution Inputs to Deep Generative Models Using Typicality*. 2019. arXiv: 1906.02994 (cited on pages 24, 28, 46, 56, 60, 63, 64, 68, 72, 73, 157, 163).

[474] Narayanan, S., Bresch, E., Ghosh, P. K., Goldstein, L., Katsamanis, A., Kim, Y., Lammert, A., Proctor, M., Ramanarayanan, V., Zhu, Y., "A Multimodal Real-Time MRI Articulatory Corpus for Speech Research". In: *Annual Conference of the International Speech Communication Association*. 2011 (cited on page 198).

[475] National Transportation Safety Board (NHSA), *Collision Between Vehicle Controlled by Developmental Automated Driving System and Pedestrian, Tempe, Arizona, March 18, 2018*. Highway Accident Report NTSB/HAR-19/03 PB2019-101402. Washington D.C., 2019 (cited on pages 6, 7).

[476] Navi, B. B., Audebert, H. J., Alexandrov, A. W., Cadilhac, D. A., Grotta, J. C., PRESTO (Prehospital Stroke Treatment Organization) Writing Group, "Mobile Stroke Units: Evidence, Gaps, and next Steps". In: *Stroke* 53.6 (2022) (cited on pages 8, 140).

[477] Nazábal, A., Olmos, P. M., Ghahramani, Z., Valera, I., "Handling Incomplete Heterogeneous Data Using VAEs". In: *Pattern Recognition* 107 (2020) (cited on page 75).

[478] Ndiour, I., Ahuja, N., Tickoo, O., *Out-of-Distribution Detection with Subspace Techniques and Probabilistic Modeling of Features*. 2020. arXiv: 2012.04250 (cited on pages 24, 27).

[479] Neal, R. M. "Bayesian Learning for Neural Networks". University of Toronto, 1995 (cited on page 13).

[480] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A. Y., "Reading Digits in Natural Images with Unsupervised Feature Learning". In: *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*. 2011 (cited on pages 50, 69, 227).

[481] Neugebauer, O., Sachs, A., Götze, A., *Mathematical Cuneiform Texts*. New Haven, Conn.: The American Oriental Society and the American Schools of Oriental Research, 1945 (cited on page 3).

[482] Neyman, J., Pearson, E. S., "On the Use and Interpretation of Certain Test Criteria for Purposes of Statistical Inference: Part I". In: *Biometrika* (1928) (cited on page 61).

[483] Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A. Y., "Multimodal Deep Learning". In: *Proceedings of the 28th International Conference on Machine Learning (ICML)*. 2011 (cited on page 198).

[484] Nguyen, A. T., Lu, F., Munoz, G. L., Raff, E., Nicholas, C., Holt, J., "Out of Distribution Data Detection Using Dropout Bayesian Neural Networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Volume 36. 7. 2022 (cited on page 27).

[485] Nguyen, A., Yosinski, J., Clune, J., "Deep Neural Networks Are Easily Fooled: High Confidence Predictions for Unrecognizable Images". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Volume 07-12-June. 2015. ISBN: 978-1-4673-6964-0 (cited on page 41).

[486] Nguyen, T. A., Seyssel, M., Rozé, P., Rivière, M., Kharitonov, E., Baevski, A., Dunbar, E., Dupoux, E., *The Zero Resource Speech Benchmark 2021: Metrics and Baselines for Unsupervised Spoken Language Modeling*. 2020. arXiv: 2011.11588. (Visited on 12 April 2023) (cited on page 222).

[487] Nguyen, T. A., Kharitonov, E., Copet, J., Adi, Y., Hsu, W.-N., Elkahky, A., Tomasello, P., Algayres, R., Sagot, B., Mohamed, A., Dupoux, E., *Generative Spoken Dialogue Language Modeling*. 2022. arXiv: 2203.16502 (cited on pages 185, 223, 226).

[488] Nguyen, T. A., Sagot, B., Dupoux, E., "Are Discrete Units Necessary for Spoken Language Modeling?" In: *IEEE Journal on Selected Topics in Signal Processing (JSTSP)* 16.6 (2022) (cited on page 226).

[489] Niculescu-Mizil, A., Caruana, R., "Predicting Good Probabilities with Supervised Learning". In: *Proceedings of the International Conference on Machine Learning (ICML)*. New York, NY, USA: Association for Computing Machinery, 2005. ISBN: 1-59593-180-5 (cited on page 11).

[490] Nissen, H. J., Damerow, P., Englund, R. K., *Archaic Bookkeeping: Early Writing and Techniques of Economic Administration in the Ancient Near East*. University of Chicago Press, 1993 (cited on pages 3, 4).

[491] Nouraei, S., Hudovsky, A., Virk, J., Chatrath, P., Sandhu, G., "An Audit of the Nature and Impact of Clinical Coding Subjectivity Variability and Error in Otolaryngology". In: *Clinical Otolaryngology* 38.6 (2013). ISSN: 1749-4486 (cited on page 138).

[492] O'Malley, K. J., Cook, K. F., Price, M. D., Wildes, K. R., Hurdle, J. F., Ashton, C. M., "Measuring Diagnoses: ICD Code Accuracy". In: *Health Services Research* 40 (5 Pt 2 2005). ISSN: 0017-9124. pmid: 16178999 (cited on page 116).

[493] O'Shaughnessy, D. "Linear Predictive Coding". In: *IEEE Potentials* 7.1 (1988) (cited on page 186).

[494] Olshausen, B. A., Field, D. J., "Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images". In: *Nature* 381.6583 (1996). ISSN: 00280836. pmid: 8637596 (cited on page 179).

[495] Ondel, L., Burget, L., Černocký, J., "Variational Inference for Acoustic Unit Discovery". In: *Procedia Computer Science*. SLTU-2016 5th Workshop on Spoken Language Technologies for Under-resourced Languages 09-12 May 2016 Yogyakarta, Indonesia 81 (2016). ISSN: 1877-0509 (cited on pages 94, 223).

[496] Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K., "WaveNet: A Generative Model for Raw Audio". In: *Proceedings of the 9th ISCA Speech Synthesis Workshop*. Sunnyvale, CA, USA, 2016 (cited on pages 30, 42, 82, 92, 97, 98, 102, 185, 221, 242).

[497] Oord, A., Kalchbrenner, N., Espeholt, L., Kavukcuoglu, K., Vinyals, O., Graves, A., "Conditional Image Generation with PixelCNN Decoders". In: *Proceedings of the 29th International Conference on Neural Information Processing Systems (NeurIPS)*. Barcelona, Spain, 2016 (cited on pages 28, 30, 45, 97).

[498] Oord, A., Kalchbrenner, N., Kavukcuoglu, K., "Pixel Recurrent Neural Networks". In: *Proceedings of the 33rd International Conference on Machine Learning (ICML)*. New York, NY, USA: Journal of Machine Learning, 2016 (cited on pages 42, 47, 97).

[499] Oord, A., Li, Y., Vinyals, O., *Representation Learning with Contrastive Predictive Coding*. 2018. arXiv: 1807.03748 (cited on pages 81, 82, 85, 91, 95, 157, 180, 190–192, 205, 206).

[500] Oord, A., Vinyals, O., Kavukcuoglu, K., "Neural Discrete Representation Learning". In: *Proceedings of the 30th Conference on Neural Information Processing Systems (NeurIPS)*. Long Beach, CA, USA, 2018 (cited on pages 84, 85, 91–93, 98, 99, 185, 190).

[501] Oord, A. "Parallel WaveNet: Fast High-Fidelity Speech Synthesis". In: *Proceedings of the 35th International Conference on Machine Learning (ICML)*. Volume 80. Proceedings of Machine Learning Research. Stockholm, Sweden: PMLR, 2018. arXiv: 1711.10433 (cited on pages 105, 242).

[502] Oostema, J. A., Carle, T., Talia, N., Reeves, M., "Dispatcher Stroke Recognition Using a Stroke Screening Tool: A Systematic Review". In: *Cerebrovascular Diseases* 42.5-6 (2016) (cited on pages 8, 140).

[503] Opitz, J., Burst, S., *Macro F1 and Macro F1*. arXiv, 2021. arXiv: 1911.03347 (cited on page 122).

[504] Oponowicz, T. "Spoken Language Identification". In: (2018) (cited on pages 204, 205).

[505] Ouali, Y., Hudelot, C., Tami, M., "An Overview of Deep Semi-Supervised Learning". 2020. arXiv: 2006.05278 (cited on page 81).

[506] Panayotov, V., Chen, G., Povey, D., Khudanpur, S., "Librispeech: An ASR Corpus Based on Public Domain Audio Books". In: *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brisbane, Australia, 2015 (cited on pages 105, 201, 242).

[507] Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., Le, Q. V., *SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition*. 2019. arXiv: 1904.08779 (cited on page 83).

[508] Park, D. S., Zhang, Y., Jia, Y., Han, W., Chiu, C.-C., Li, B., Wu, Y., Le, Q. V., "Improved Noisy Student Training for Automatic Speech Recognition". In: *Annual Conference of the International Speech Communication Association*. 2020 (cited on pages 181, 209, 210).

[509] Park, H., Castaño, J., Ávila, P., Pérez, D., Berinsky, H., Gambarte, L., Luna, D., Otero, C., "An Information Retrieval Approach to ICD-10 Classification". In: *Studies in Health Technology and Informatics* 264 (2019). ISSN: 1879-8365. pmid: 31438233 (cited on page 116).

[510] Parthasarathi, S. H. K., Strom, N., *Lessons from Building Acoustic Models with a Million Hours of Speech*. 2019. arXiv: 1904.01624 (cited on page 181).

[511] Parzen, E. "On Estimation of a Probability Density Function and Mode". In: *The Annals of Mathematical Statistics* 33.3 (1962) (cited on page 28).

[512] Pasad, A., Shi, B., Kamper, H., Livescu, K., "On the Contributions of Visual and Textual Supervision in Low-Resource Semantic Speech Retrieval". In: *Annual Conference of the International Speech Communication Association* (2019) (cited on page 199).

[513] Pasad, A., Chou, J.-C., Livescu, K., "Layer-Wise Analysis of a Self-Supervised Speech Representation Model". In: *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. 2021 (cited on pages 162, 169, 183, 212, 213).

[514] Pasad, A., Wu, F., Shon, S., Livescu, K., Han, K. J., *On the Use of External Data for Spoken Named Entity Recognition*. 2022. arXiv: 2112.07648. (Visited on 23 September 2023) (cited on page 95).

[515] Pascual, D., Luck, S., Wattenhofer, R., "Towards BERT-based Automatic ICD Coding: Limitations and Opportunities". In: *Proceedings of the 20th Workshop on Biomedical Language Processing*. Online: Association for Computational Linguistics, 2021 (cited on pages 116, 137, 171).

[516] Pascual, S., Ravanelli, M., Serrà, J., Bonafonte, A., Bengio, Y., "Learning Problem-Agnostic Speech Representations from Multiple Self-Supervised Tasks". In: *Annual Conference of the International Speech Communication Association*. 2019. arXiv: 1904.03416 (cited on pages 89, 95, 184, 188, 190, 206).

[517] Paszke, A., Chanan, G., Lin, Z., Gross, S., Yang, E., Antiga, L., Devito, Z., "Automatic Differentiation in PyTorch". In: *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS)*. 2017 (cited on pages 164, 197, 244).

[518] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A. A., "Context Encoders: Feature Learning by Inpainting". In: 2016 (cited on pages 38, 79).

[519] Paul, D. B., Baker, J., "The Design for the Wall Street Journal-based CSR Corpus". In: *Proceedings of the Worhshop on Speech and Natural Language:* Harriman, New York, 1992 (cited on page 201).

[520]   Pearce, D., Hirsch, H.-G., "The Aurora Experimental Framework for the Performance Evalua-
        tion of Speech Recognition Systems under Noisy Conditions". In: *6th International Conference
        on Spoken Language Processing (ICSLP 2000)*. ISCA, 2000 (cited on page 163).

[521]   Peng, P., Harwath, D., "Fast-Slow Transformer for Visually Grounding Speech". In: *IEEE
        International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022 (cited on
        page 199).

[522]   Peng, P., Harwath, D., "Self-Supervised Representation Learning for Speech Using Visual
        Grounding and Masked Language Modeling". In: *AAAI SAS Workshop*. 2022 (cited on page 199).

[523]   Peng, P., Harwath, D., *Word Discovery in Visually Grounded, Self-Supervised Speech Models*. 2022.
        arXiv: 2203.15081 (cited on page 199).

[524]   Peng, P., Kamper, H., Livescu, K., "A Correspondence Variational Autoencoder for Unsuper-
        vised Acoustic Word Embeddings". In: *Procdings of the NeurIPS Workshop on Self-Supervised
        Learning for Speech and Audio Processing*. 2020 (cited on page 200).

[525]   Perronnin, F., Sánchez, J., Mensink, T., "Improving the Fisher Kernel for Large-Scale Image
        Classification". In: *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision,
        Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*. Springer, 2010 (cited on
        page 262).

[526]   Petajan, E. D. "Automatic Lipreading to Enhance Speech Recognition (Speech Reading)". PhD
        thesis. University of Illinois at Urbana-Champaign, USA, 1984 (cited on page 197).

[527]   Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., *Deep
        Contextualized Word Representations*. 2018. arXiv: 1802.05365 (cited on pages 180, 187).

[528]   Platt, J. "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized
        Likelihood Methods". In: *Advances in large margin classifiers* 10.3 (1999) (cited on pages 11, 165).

[529]   Polyak, A., Adi, Y., Copet, J., Kharitonov, E., Lakhotia, K., Hsu, W.-N., Mohamed, A., Dupoux,
        E., "Speech Resynthesis from Discrete Disentangled Self-Supervised Representations". In:
        Annual Conference of the International Speech Communication Association. 2021 (cited on pages 185,
        223).

[530]   Potamianos, G., Neti, C., Gravier, G., Garg, A., Senior, A. W., "Recent Advances in the Au-
        tomatic Recognition of Audiovisual Speech". In: *Proceedings of the IEEE* 91.9 (2003) (cited on
        page 197).

[531]   Pratap, V., Xu, Q., Sriram, A., Synnaeve, G., Collobert, R., "MLS: A Large-Scale Multilingual
        Dataset for Speech Research". In: *Annual Conference of the International Speech Communication
        Association*. 2020 (cited on page 202).

[532]   Primewords Information Technology Co., Ltd. *Primewords Chinese Corpus Set 1*. 2018 (cited on
        page 203).

[533]   Pu, J., Yang, Y., Li, R., Elibol, O., Droppo, J., "Scaling Effect of Self-Supervised Models". In:
        *Proceedings of the 22nd Annual Conference of the International Speech Communication Association
        (Interspeech)*. ISCA, 2021 (cited on page 215).

[534]   Qian, K., Zhang, Y., Gao, H., Ni, J., Lai, C.-I., Cox, D., Hasegawa-Johnson, M., Chang, S.,
        "ContentVec: An Improved Self-Supervised Speech Representation by Disentangling Speak-
        ers". In: *International Conference on Machine Learning (ICML)*. 2022 (cited on page 225).

[535]   Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., Huang, X., "Pre-Trained Models for Natural Language
        Processing: A Survey". In: *Science China Technological Sciences* 63.10 (2020). ISSN: 1869-1900
        (cited on page 178).

[536]   Quitry, F. d. C., Tagliasacchi, M., Roblek, D., "Learning Audio Representations via Phase Pre-
        diction". 2019. arXiv: 1910.11910 (cited on pages 188, 204–206).

[537] Rabiner, L., Wilpon, J., "Considerations in Applying Clustering Techniques to Speaker Independent Word Recognition". In: *Ieee International Conference on Acoustics, Speech, and Signal Processing*. Volume 4. 1979 (cited on page 178).

[538] Rabiner, L. R. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". In: *Proceedings of the IEEE* 77.2 (1989) (cited on page 31).

[539] Radford, A., Kim, J. W., Xu, T., Brockman, G., Mcleavey, C., Sutskever, I., "Robust Speech Recognition via Large-Scale Weak Supervision". In: *Proceedings of the International Conference on Machine Learning*. Volume 202. Proceedings of Machine Learning Research. PMLR, 2023 (cited on page 147).

[540] Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., "Improving Language Understanding by Generative Pre-Training". In: (2018) (cited on page 30).

[541] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., "Language Models Are Unsupervised Multitask Learners". In: *OpenAI blog* 1.8 (2019) (cited on page 180).

[542] Rainforth, T., Kosiorek, A. R., Le, T. A., Maddison, C. J., Igl, M., Wood, F., Teh, Y. W., *Tighter Variational Bounds Are Not Necessarily Better*. 2019. arXiv: 1802.04537 (cited on pages 36, 161, 170).

[543] Ranganath, R., Tran, D., Blei, D. M., *Hierarchical Variational Models*. 2016. arXiv: 1511.02386 (cited on page 30).

[544] Ranzato, M., Boureau, Y.-L., Chopra, S., LeCun, Y., "A Unified Energy-Based Framework for Unsupervised Learning". In: *Procedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2007 (cited on page 179).

[545] Rao, C. R. "Large Sample Tests of Statistical Hypotheses Concerning Several Parameters with Applications to Problems of Estimation". In: *Mathematical Proceedings of the Cambridge Philosophical Society*. Volume 44. 1. Cambridge University Press, 1948 (cited on pages 29, 61).

[546] Rao, C. R. "Score Test: Historical Review and Recent Developments". In: *Advances in ranking and selection, multiple comparisons, and reliability: methodology and applications* (2005) (cited on page 61).

[547] Ravanelli, M., Bengio, Y., *Learning Speaker Representations with Mutual Information*. 2018. arXiv: 1812.00271 (cited on pages 206, 207).

[548] Ravanelli, M., Cristoforetti, L., Gretter, R., Pellin, M., Sosi, A., Omologo, M., "The DIRHA-English Corpus and Related Tasks for Distant-Speech Recognition in Domestic Environments". In: *IEEE Workshop on Automatic Speech Recognition and Understanding*. 2015 (cited on page 203).

[549] Ravanelli, M., Zhong, J., Pascual, S., Swietojanski, P., Monteiro, J., Trmal, J., Bengio, Y., "Multi-Task Self-Supervised Learning for Robust Speech Recognition". In: *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020 (cited on pages 89, 188, 190, 206).

[550] Ravi, S., Knight, K., "Deciphering Foreign Language". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL)*. Portland, Oregon, USA: Association for Computational Linguistics, 2011 (cited on page 218).

[551] Razavi, A., Oord, A., Vinyals, O., *Generating Diverse High-Fidelity Images with VQ-VAE-2*. 2019. arXiv: 1906.00446 (cited on page 193).

[552] Ren, J., Liu, P. J., Fertig, E., Snoek, J., Poplin, R., Depristo, M., Dillon, J., Lakshminarayanan, B., "Likelihood Ratios for Out-of-Distribution Detection". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS)*. Vancouver, Canada, 2019 (cited on pages 24–26, 28, 42, 46, 47, 51, 53, 56, 62, 68, 72).

[553]   Renduchintala, A., Ding, S., Wiesner, M., Watanabe, S., "Multi-Modal Data Augmentation for End-to-End ASR". In: *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*. Hyderabad, India: ISCA, 2018 (cited on page 221).

[554]   Renn, J. "Learning from Kushim About the Origins of Writing and Farming". In: *Culture and Cognition: Essays in Honor of Peter Damerow*. Berlin: Edition Open Access, Max Planck Institute for the History of Science, 2019. ISBN: 978-3-945561-35-5 (cited on page 4).

[555]   Renshaw, D., Kamper, H., Jansen, A., Goldwater, S., "A Comparison of Neural Network Methods for Unsupervised Representation Learning on the Zero Resource Speech Challenge". In: *Annual Conference of the International Speech Communication Association* (2015) (cited on pages 88, 186).

[556]   Rezende, D. J., Mohamed, S., "Variational Inference with Normalizing Flows". In: *Proceedings of the 32nd International Conference on Machine Learning (ICML)*. Lille, France, 2015 (cited on pages 28, 30, 45, 99).

[557]   Rezende, D. J., Mohamed, S., Wierstra, D., "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". In: *Proceedings of the 31st International Conference on Machine Learning (ICML)*. Volume 32. Beijing, China: PMLR, 2014 (cited on pages 28, 30, 31, 35, 38, 42, 45, 61, 69, 90, 95, 97, 162, 180, 185).

[558]   Rivière, M., Joulin, A., Mazaré, P.-E., Dupoux, E., *Unsupervised Pretraining Transfers Well across Languages*. 2020. arXiv: 2002.02848. (Visited on 29 October 2021) (cited on pages 89, 190, 206, 208, 215, 216).

[559]   Rizzo, S. G., Montesi, D., Fabbri, A., Marchesini, G., "ICD Code Retrieval: Novel Approach for Assisted Disease Classification". In: *Data Integration in the Life Sciences*. Volume 9162. Cham: Springer International Publishing, 2015. ISBN: 978-3-319-21842-7 978-3-319-21843-4 (cited on page 116).

[560]   Rochette, G., Manoel, A., Tramel, E. W., *Efficient Per-Example Gradient Computations in Convolutional Neural Networks*. 2019. arXiv: 1912.06015 (cited on page 66).

[561]   Rodríguez-Fuentes, L. J., Varona, A., Penagarikano, M., Bordel, G., Diez, M., "GTTS-EHU Systems for QUESST at MediaEval 2014". In: *MediaEval*. 2014 (cited on page 211).

[562]   Roeder, G., Wu, Y., Duvenaud, D., *Sticking the Landing: Simple, Lower-Variance Gradient Estimators for Variational Inference*. 2017. arXiv: 1703.09194 (cited on pages 36, 161, 170).

[563]   Rogers, A., Kovaleva, O., Rumshisky, A., "A Primer in BERTology: What We Know about How BERT Works". In: *Transactions of the Association for Computational Linguistics (ACL)* 8 (2020) (cited on page 178).

[564]   Rosenberg, A., Zhang, Y., Ramabhadran, B., Jia, Y., Moreno, P., Wu, Y., Wu, Z., "Speech Recognition with Augmented Synthesized Speech". In: *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. 2019 (cited on page 221).

[565]   Rosenblatt, F. "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain." In: *Psychological Review* 65 (1958) (cited on page 150).

[566]   Rouditchenko, A., Boggust, A., Harwath, D., Chen, B., Joshi, D., Thomas, S., Audhkhasi, K., Kuehne, H., Panda, R., Feris, R., "AVLnet: Learning Audio-Visual Language Representations from Instructional Videos". In: *Annual Conference of the International Speech Communication Association*. 2021 (cited on page 199).

[567]   Rousseau, A., Deléglise, P., Esteve, Y., "TED-LIUM: An Automatic Speech Recognition Dedicated Corpus". In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*. 2012 (cited on page 202).

[568] Roy, D. "Learning from Sights and Sounds: A Computational Model". PhD thesis. MIT Media Laboratory, USA, 1999 (cited on page 198).

[569] Ruff, L., Vandermeulen, R. A., Görnitz, N., Binder, A., Müller, E., Müller, K.-R., Kloft, M., *Deep Semi-Supervised Anomaly Detection*. 2020. arXiv: 1906.02694. (Visited on 12 September 2023) (cited on page 24).

[570] Sadhu, S., He, D., Huang, C.-W., Mallidi, S. H., Wu, M., Rastrow, A., Stolcke, A., Droppo, J., Maas, R., *Wav2vec-C: A Self-supervised Model for Speech Representation Learning*. 2021. arXiv: 2103.08393 (cited on pages 190, 193, 201, 203, 207, 208).

[571] Sakurada, M., Yairi, T., "Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction". In: *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis (MLSDA)*. Gold Coast, QLD, Austrailia, 2014 (cited on pages 24, 29).

[572] Salakhutdinov, R., Larochelle, H., "Efficient Learning of Deep Boltzmann Machines". In: *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*. JMLR Workshop and Conference Proceedings, 2010 (cited on page 30).

[573] Salimans, T., Karpathy, A., Chen, X., Kingma, D. P., "PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications". In: *Proceedings of the 5th International Conference on Learning Representations (ICLR)*. Toulon, France, 2017 (cited on pages 42, 46, 47, 51, 69, 97, 105, 228).

[574] Salimans, T., Kingma, D. P., "Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks". In: *Proceedings of the 30th Conference on Neural Information Processing Systems (NeurIPS)*. Barcelona, Spain, 2016 (cited on pages 51, 228, 230, 264).

[575] Sanabria, R., Waters, A., Baldridge, J., "Talk, Don't Write: A Study of Direct Speech-Based Image Retrieval". In: *Annual Conference of the International Speech Communication Association*. 2021 (cited on page 199).

[576] Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J., "Image Classification with the Fisher Vector: Theory and Practice". In: *International journal of computer vision* 105 (2013) (cited on page 262).

[577] Saxena, V., Ba, J., Hafner, D., "Clockwork Variational Autoencoders". In: *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*. Virtual, 2021. arXiv: 2102.09532 (cited on pages 17, 100, 103, 160, 170, 243, 251, 253).

[578] Scharenborg, O. "Linguistic Unit Discovery from Multi-Modal Inputs in Unwritten Languages: Summary of the "Speaking Rosetta" JSALT 2017 Workshop". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018 (cited on page 199).

[579] Schatz, T., Peddinti, V., Bach, F., Jansen, A., Hermansky, H., Dupoux, E., "Evaluating Speech Features with the Minimal-Pair ABX Task: Analysis of the Classical MFC/PLP Pipeline". In: *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)* (2013) (cited on page 95).

[580] Schatz, T., Peddinti, V., Cao, X.-N., Bach, F., Hermansky, H., Dupoux, E., "Evaluating Speech Features with the Minimal-Pair ABX Task (II): Resistance to Noise". In: *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)* (2014) (cited on page 95).

[581] Schirrmeister, R., Zhou, Y., Ball, T., Zhang, D., "Understanding Anomaly Detection with Deep Invertible Networks through Hierarchies of Distributions and Features". In: *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*. Virtual, 2020 (cited on pages 43, 62, 68).

[582] Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U., Langs, G., "Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery". In: *International Conference on Information Processing in Medical Imaging (IPMI)*. Boone, NC, USA: Springer, 2017 (cited on pages 24, 29).

[583] Schmidt, M., Pedersen, L., Sørensen, H. T., "The Danish Civil Registration System as a Tool in Epidemiology". In: *European Journal of Epidemiology* 29 (2014) (cited on page 148).

[584] Schneider, S., Baevski, A., Collobert, R., Auli, M., "Wav2vec: Unsupervised Pre-training for Speech Recognition". In: *Proceedings of the 20th Annual Conference of the International Speech Communication Association (Interspeech)*. Graz, Austria: ISCA, 2019. arXiv: 1904.05862 (cited on pages 28, 38, 81, 83, 85, 95, 190, 192, 206).

[585] Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., Williamson, R. C., "Estimating the Support of a High-Dimensional Distribution". In: *Neural Computation* 13.7 (2001) (cited on page 28).

[586] Schultz, M., Joachims, T., "Learning a Distance Metric from Relative Comparisons". In: *Advances in Neural Information Processing Systems*. 2003 (cited on page 191).

[587] Seabold, S., Perktold, J., "Statsmodels: Econometric and Statistical Modeling with Python". In: *9th Python in Science Conference*. 2010 (cited on page 275).

[588] Searle, T., Ibrahim, Z., Dobson, R., "Experimental Evaluation and Development of a Silver-Standard for the MIMIC-III Clinical Coding Dataset". In: *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*. Online: Association for Computational Linguistics, 2020 (cited on pages 138, 171).

[589] Sechidis, K., Tsoumakas, G., Vlahavas, I., "On the Stratification of Multi-label Data". In: *Machine Learning and Knowledge Discovery in Databases*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2011. ISBN: 978-3-642-23808-6 (cited on pages 122, 125).

[590] Sehwag, V., Chiang, M., Mittal, P., "SSD: A Unified Framework for Self-Supervised Outlier Detection". In: *International Conference on Learning Representations*. Virtual, 2021 (cited on pages 24, 30).

[591] Sennrich, R., Haddow, B., Birch, A., "Improving Neural Machine Translation Models with Monolingual Data". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2016 (cited on page 221).

[592] Serrà, J., Álvarez, D., Gómez, V., Slizovskaia, O., Núñez, J. F., Luque, J., "Input Complexity and Out-of-Distribution Detection with Likelihood-Based Generative Models". In: *Proceedings of the 8th International Conference on Learning Representations (ICLR)*. Addis Ababa, Ethiopia, 2020 (cited on pages 24, 28, 46, 47, 52, 53, 57, 62, 68).

[593] Settle, S., Audhkhasi, K., Livescu, K., Picheny, M., "Acoustically Grounded Word Embeddings for Improved Acoustics-to-Word Speech Recognition". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019 (cited on pages 186, 200).

[594] Settle, S., Levin, K., Kamper, H., Livescu, K., "Query-by-Example Search with Discriminative Neural Acoustic Word Embeddings". In: *Annual Conference of the International Speech Communication Association*. 2017 (cited on page 199).

[595] Shailaja, K., Seetharamulu, B., Jabbar, M. A., "Machine Learning in Healthcare: A Review". In: *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. IEEE, 2018 (cited on page 3).

[596] Shannon, C. E. "A Mathematical Theory of Communication". In: *The Bell System Technical Journal* 27 (July 1948 1948). ISSN: 07246811. pmid: 9230594 (cited on pages 19, 20, 57, 106).

[597] Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerry-Ryan, R. J., Saurous, R. A., Agiomyrgiannakis, Y., Wu, Y., *Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions*. 2018. arXiv: 1712.05884 (cited on page 221).

[598] Shi, B., Hsu, W.-N., Lakhotia, K., Mohamed, A., *Learning Audio-Visual Speech Representation by Masked Multimodal Cluster Prediction*. 2022. arXiv: 2201.02184 (cited on page 198).

[599] Shi, B., Hsu, W.-N., Mohamed, A., *Robust Self-Supervised Audio-Visual Speech Recognition*. 2022. arXiv: 2201.01763 (cited on page 198).

[600] Shi, H., Xie, P., Hu, Z., Zhang, M., Xing, E. P., "Towards Automated ICD Coding Using Deep Learning". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2018. arXiv: 1711.04075 (cited on page 117).

[601] Shi, J., Chang, X., Hayashi, T., Lu, Y.-J., Watanabe, S., Xu, B., *Discretization and Re-Synthesis: An Alternative Method to Solve the Cocktail Party Problem*. 2022. arXiv: 2112.09382. (Visited on 12 April 2023) (cited on page 224).

[602] Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., Catanzaro, B., *Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism*. 2020. arXiv: 1909.08053 (cited on page 180).

[603] Shon, S., Pasad, A., Wu, F., Brusco, P., Artzi, Y., Livescu, K., Han, K. J., "SLUE: New Benchmark Tasks for Spoken Language Understanding Evaluation on Natural Speech". 2021. arXiv: 2111.10367 (cited on pages 95, 157).

[604] Shor, J., Jansen, A., Maor, R., Lang, O., Tuval, O., Chaumont Quitry, F., Tagliasacchi, M., Shavitt, I., Emanuel, D., Haviv, Y., "Towards Learning a Universal Non-Semantic Representation of Speech". In: *Annual Conference of the International Speech Communication Association*. 2020 (cited on pages 211, 212).

[605] Siddiqui, F., Merril, J. B., "17 Fatalities, 736 Crashes; The Shocking Toll of Tesla's Autopilot". In: *The Washington Post. Tech* (2023) (cited on page 7).

[606] Simonyan, K., Zisserman, A., *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. arXiv: 1409.1556 (cited on page 79).

[607] Sinha, S., Dieng, A. B., *Consistency Regularization for Variational Auto-Encoders*. 2021. arXiv: 2105.14859 (cited on page 97).

[608] Sivaram, G. S., Nemala, S. K., Elhilali, M., Tran, T. D., Hermansky, H., "Sparse Coding for Speech Recognition". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2010 (cited on page 179).

[609] Smith, N., Gales, M., "Speech Recognition Using SVMs". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2001 (cited on page 179).

[610] Smolensky, P. "Chapter 6: Information Processing in Dynamical Systems: Foundations of Harmony Theory". In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*. MIT Press, 1986 (cited on page 94).

[611] Snyder, D., Chen, G., Povey, D., *MUSAN: A Music, Speech, and Noise Corpus*. 2015. arXiv: 1510.08484 (cited on pages 204, 205).

[612] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S., "Deep Unsupervised Learning Using Nonequilibrium Thermodynamics". In: *International Conference on Machine Learning (ICML)*. 2015 (cited on page 30).

[613] Sønderby, C. K., Caballero, J., Theis, L., Shi, W., Huszár, F., "Amortised MAP Inference for Image Super-Resolution". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. Toulon, France, 2017 (cited on page 38).

[614] Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., Winther, O., "Ladder Variational Autoencoders". In: *Proceedings of the 29th Conference on Neural Information Processing Systems (NeurIPS)*. Barcelona, Spain, 2016 (cited on pages 36, 45, 46, 92, 97, 99, 101, 103–105, 159, 228, 235, 252).

[615] Song, X., Wang, G., Huang, Y., Wu, Z., Su, D., Meng, H., "Speech-XLNet: Unsupervised Acoustic Model Pretraining for Self-Attention Networks". In: *Proceedings of the 21st Annual Conference of the International Speech Communication Association (Interspeech)*. ISCA, 2020 (cited on pages 188, 190, 202, 207).

[616] Song, Y., Ermon, S., "Generative Modeling by Estimating Gradients of the Data Distribution". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems (NeurIPS)*. Vancouver, BC, Canada, 2019 (cited on page 30).

[617] Soong, F., Rosenberg, A., Juang, L. R. B., "A Vector Quantization Approach to Speaker Recognition". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (1985) (cited on pages 87, 185).

[618] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. R., "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15 (2014) (cited on page 188).

[619] Srivastava, N., Salakhutdinov, R. R., "Multimodal Learning with Deep Boltzmann Machines". In: *Conference on Neural Information Processing Systems (Neurips)*. 2012 (cited on page 198).

[620] Stanfill, M. H., Williams, M., Fenton, S. H., Jenders, R. A., Hersh, W. R., "A Systematic Literature Review of Automated Clinical Coding and Classification Systems". In: *Journal of the American Medical Informatics Association: JAMIA* 17.6 (2010). ISSN: 1527-974X. pmid: 20962126 (cited on page 116).

[621] Stowell, D., Wood, M. D., Pamuła, H., Stylianou, Y., Glotin, H., "Automatic Acoustic Detection of Birds through Deep Learning: The First Bird Audio Detection Challenge". In: *Methods in Ecology and Evolution* 10.3 (2019) (cited on pages 204, 205).

[622] Sun, Y., Guo, C., Li, Y., "React: Out-of-distribution Detection with Rectified Activations". In: *Advances in Neural Information Processing Systems* 34 (2021) (cited on page 27).

[623] Surfingtech Co., Ltd. *ST-CMDS: Free ST Chinese Mandarin Corpus* (cited on page 203).

[624] Synnaeve, G., Versteegh, M., Dupoux, E., "Learning Words from Images and Speech". In: *Conference on Neural Information Processing Systems (NeurIPS), Workshop on Learning Semantics*. 2014 (cited on page 199).

[625] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., "Going Deeper with Convolutions". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015 (cited on page 79).

[626] Tachbelie, M. Y., Abate, S. T., Besacier, L., "Using Different Acoustic, Lexical and Language Modeling Units for ASR of an under-Resourced Language–Amharic". In: *Speech Communication* 56 (2014) (cited on page 204).

[627] Tack, J., Mo, S., Jeong, J., Shin, J., "CSI: Novelty Detection via Contrastive Learning on Distributionally Shifted Instances". In: *Advances in Neural Information Processing Systems* 33 (2020) (cited on pages 24, 30).

[628] Tagliasacchi, M., Gfeller, B., Quitry, F. d. C., Roblek, D., "Pre-Training Audio Representations with Self-Supervision". In: *IEEE Signal Processing Letters* 27 (2020). ISSN: 1558-2361 (cited on pages 88, 188, 190, 204–206).

[629] Tagliasacchi, M., Gfeller, B., Quitry, F. d. C., Roblek, D., *Self-Supervised Audio Representation Learning for Mobile Devices*. 2019. arXiv: 1905.11796 (cited on page 188).

[630] Talnikar, C., Likhomanenko, T., Collobert, R., Synnaeve, G., "Joint Masked CPC and CTC Training for ASR". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021 (cited on page 89).

[631] Tanaka, M., Torii, A., Okutomi, M., "Fisher Vector Based on Full-Covariance Gaussian Mixture Model". In: *Information and Media Technologies* 8.4 (2013) (cited on page 262).

[632] Tay, Y., Dehghani, M., Bahri, D., Metzler, D., *Efficient Transformers: A Survey*. 2022. arXiv: 2009. 06732. (Visited on 5 April 2022) (cited on page 225).

[633] Teng, F., Liu, Y., Li, T., Zhang, Y., Li, S., Zhao, Y., "A Review on Deep Neural Networks for ICD Coding". In: *IEEE Transactions on Knowledge and Data Engineering* (2022). ISSN: 1041-4347, 1558-2191, 2326-3865 (cited on pages 116, 117, 119, 121, 137, 171).

[634] Teng, F., Yang, W., Chen, L., Huang, L., Xu, Q., "Explainable Prediction of Medical Codes With Knowledge Graphs". In: *Frontiers in Bioengineering and Biotechnology* (2020) (cited on pages 116, 124, 137).

[635] Terrell, G. R. "The Gradient Statistic". In: *Comput. Sci. Stat.* 34 (2002) (cited on page 61).

[636] Theis, L., Oord, A., Bethge, M., "A Note on the Evaluation of Generative Models". In: *Proceedings of the 4th International Conference on Learning Representations (ICLR)*. San Juan, Puerto Rico, 2016 (cited on pages 54, 105).

[637] Thomas, B., Kessler, S., Karout, S., "Efficient Adapter Transfer of Self-Supervised Speech Models for Automatic Speech Recognition". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022 (cited on page 224).

[638] Tieleman, T., Hinton, G., "Lecture 6.5-RMSProp: Divide the Gradient by a Running Average of Its Recent Magnitude". In: *Coursera: Neural Networks for Machine Learning* (2012) (cited on page 259).

[639] Tipping, M. E., Bishop, C. M., "Probabilistic Principal Component Analysis". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 61.3 (1999) (cited on page 31).

[640] Tjandra, A., Sakti, S., Nakamura, S., "End-to-End Feedback Loss in Speech Chain Framework via Straight-through Estimator". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019 (cited on page 221).

[641] Tjandra, A., Sakti, S., Nakamura, S., "Listening While Speaking: Speech Chain by Deep Learning". In: *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. 2017 (cited on page 220).

[642] Tjandra, A., Sakti, S., Nakamura, S., "Machine Speech Chain with One-Shot Speaker Adaptation". In: *Proceedings of the 19th Annual Conference of the International Speech Communication Association (Interspeech)*. ISCA, 2018 (cited on page 220).

[643] Tjandra, A., Sakti, S., Nakamura, S., "Transformer VQ-VAE for Unsupervised Unit Discovery and Speech Synthesis: ZeroSpeech 2020 Challenge". In: *Annual Conference of the International Speech Communication Association*. 2020 (cited on page 212).

[644] Tjandra, A., Sisman, B., Zhang, M., Sakti, S., Li, H., Nakamura, S., "VQVAE Unsupervised Unit Discovery and Multi-Scale Code2Spec Inverter for ZeroSpeech Challenge 2019". In: *IEEE Annual Conference of the International Speech Communication Association (Interspeech)*. 2019 (cited on page 223).

[645] Tomczak, J. *Trouble in Paradise: Does It Make Sense to Train Latent Variable Models with Variational Inference?* 2022. URL: https://jmtomczak.github.io/blog/13/13_trouble_in_paradise. html (cited on pages 37, 159).

[646] Tonekaboni, S., Joshi, S., McCradden, M. D., Goldenberg, A., "What Clinicians Want: Contextualizing Explainable Machine Learning for Clinical End Use". In: *Proceedings of the 4th Machine Learning for Healthcare Conference*. PMLR, 2019 (cited on pages 166, 167).

[647] Toshniwal, S., Shi, H., Shi, B., Gao, L., Livescu, K., Gimpel, K., "A Cross-Task Analysis of Text Span Representations". In: *Proceedings of the 5th Workshop on Representation Learning for NLP*. 2020 (cited on page 200).

[648] Townsend, J., Bird, T., Barber, D., "Practical Lossless Compression With Latent Variables Using Bits Back Coding". In: *Proceedings of the 7th International Conference on Learning Representations (ICLR)*. New Orleans, LA, USA, 2019 (cited on pages 57, 106).

[649] Tsai, Y.-H. H., Wu, Y., Salakhutdinov, R., Morency, L.-P., "Self-Supervised Learning from a Multi-view Perspective". 2021. arXiv: `2006.05576` (cited on page 81).

[650] Tseng, P., Kaplan, R. S., Richman, B. D., Shah, M. A., Schulman, K. A., "Administrative Costs Associated With Physician Billing and Insurance-Related Activities at an Academic Health Care System". In: *JAMA : the journal of the American Medical Association* 319.7 (2018). ISSN: 0098-7484 (cited on pages 116, 137).

[651] Tucker, G., Lawson, D., Gu, S., Maddison, C. J., "Doubly Reparameterized Gradient Estimators for Monte Carlo Objectives". In: *Proceedings of the 7th International Conference on Learning Representations (ICLR)*. New Orleans, LA, USA: OpenReview.net, 2019 (cited on pages 36, 161, 170).

[652] Turc, G., Bhogal, P., Fischer, U., Khatri, P., Lobotesis, K., Mazighi, M., Schellinger, P. D., Toni, D., De Vries, J., White, P., "European Stroke Organisation (ESO)-European Society for Minimally Invasive Neurological Therapy (ESMINT) Guidelines on Mechanical Thrombectomy in Acute Ischemic Stroke". In: *Journal of Neurointerventional Surgery* 11.8 (2019) (cited on pages 8, 139).

[653] Turian, J. "HEAR: Holistic Evaluation of Audio Representations". In: *Proceedings of Machine Learning Research (PMLR): NeurIPS 2021 Competition Track*. Volume 176. 2022 (cited on pages 211, 212).

[654] Ueno, S., Mimura, M., Sakai, S., Kawahara, T., "Multi-Speaker Sequence-to-Sequence Speech Synthesis for Data Augmentation in Acoustic-to-Word Speech Recognition". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019 (cited on page 221).

[655] Umscheid, C. A., Betesh, J., VanZandbergen, C., Hanish, A., Tait, G., Mikkelsen, M. E., French, B., Fuchs, B. D., "Development, Implementation, and Impact of an Automated Early Warning and Response System for Sepsis". In: *Journal of hospital medicine* 10.1 (2015) (cited on page 166).

[656] US Army Research Laboratory (ARL) Technical Library, *Female Programmers of the ENIAC*. 1940–1945 (cited on page 5).

[657] Vahdat, A., Kautz, J., "NVAE: A Deep Hierarchical Variational Autoencoder". In: *34th Conference on Neural Information Processing Systems (NeurIPS)*. Virtual, 2020 (cited on pages 30, 37, 46, 47, 55, 97, 99, 105, 159–162, 170, 228, 235, 250).

[658] Vahdat, A., Kreis, K., Kautz, J., "Score-Based Generative Modeling in Latent Space". In: *Advances in Neural Information Processing Systems*. Volume 34. Curran Associates, Inc., 2021 (cited on page 30).

[659] Valk, J., Alumäe, T., "VoxLingua107: A Dataset for Spoken Language Recognition". In: *Proceedings of the IEEE Spoken Language Technology Workshop*. 2021 (cited on page 202).

[660] Niekerk, B., Nortje, L., Baas, M., Kamper, H., "Analyzing Speaker Information in Self-Supervised Models to Improve Zero-Resource Speech Processing". In: *Proceedings of the 22nd Annual Conference of the International Speech Communication Association (Interspeech)*. ISCA, 2021 (cited on page 213).

[661] Niekerk, B., Nortje, L., Kamper, H., "Vector-Quantized Neural Networks for Acoustic Unit Discovery in the ZeroSpeech 2020 Challenge". In: *Annual Conference of the International Speech Communication Association*. 2020 (cited on pages 87, 212, 223).

[662] Staden, L., Kamper, H., "A Comparison of Self-Supervised Speech Representations as Input Features for Unsupervised Acoustic Word Embeddings". In: *2021 IEEE Spoken Language Technology Workshop (SLT)*. 2021 (cited on page 200).

[663] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I., "Attention Is All You Need". In: *Advances in Neural Information Processing Systems*. 2017. ISBN: 978-1-57735-738-4. pmid: 1000303116 (cited on page 84).

[664] Veaux, C., Yamagishi, J., MacDonald, K., "CSTR VCTK Corpus: English Multi-Speaker Corpus for CSTR Voice Cloning Toolkit". In: (2016) (cited on page 203).

[665] Venkataramani, V., Chakrabartty, S., Byrne, W., "Support Vector Machines for Segmental Minimum Bayes Risk Decoding of Continuous Speech". In: *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. 2003 (cited on page 179).

[666] Venkatesh, K. P., Raza, M. M., Kvedar, J. C., "Automating the Overburdened Clinical Coding System: Challenges and next Steps". In: *npj Digital Medicine* 6.1 (2023). ISSN: 2398-6352 (cited on pages 116, 137, 171).

[667] Versteegh, M., Thiolliere, R., Schatz, T., Cao, X. N., Anguera, X., Jansen, A., Dupoux, E., "The Zero Resource Speech Challenge 2015". In: *IEEE Annual Conference of the International Speech Communication Association (Interspeech)*. 2015 (cited on pages 95, 215, 222).

[668] Viereck, S., Møller, T. P., Iversen, H. K., Christensen, H., Lippert, F., "Medical Dispatchers Recognise Substantial Amount of Acute Stroke during Emergency Calls". In: *Scandinavian Journal of Trauma, Resuscitation and Emergency Medicine* 24 (2016) (cited on pages 8, 140).

[669] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion". In: *Journal of Machine Learning Research* (2010) (cited on pages 179, 185).

[670] Voita, E., Sennrich, R., Titov, I., "The Bottom-up Evolution of Representations in the Transformer: A Study with Machine Translation and Language Modeling Objectives". In: *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*. 2019 (cited on page 213).

[671] Vu, T., Nguyen, D. Q., Nguyen, A., "A Label Attention Model for ICD Coding from Clinical Text". In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. Yokohama, Japan: International Joint Conferences on Artificial Intelligence Organization, 2020. ISBN: 978-0-9992411-6-5 (cited on pages 116, 119–122, 137, 171).

[672] Vyas, A., Jammalamadaka, N., Zhu, X., Das, D., Kaul, B., Willke, T. L., "Out-of-Distribution Detection Using an Ensemble of Self Supervised Leave-out Classifiers". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018. arXiv: 1809.03576 (cited on page 25).

[673] Wald, A. "Tests of Statistical Hypotheses Concerning Several Parameters When the Number of Observations Is Large". In: *Transactions of the American Mathematical society* 54.3 (1943) (cited on page 61).

[674] Wan, V., Renals, S., "SVMSVM: Support Vector Machine Speaker Verification Methodology". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2003 (cited on page 179).

[675] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S. R., *GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding*. 2018. arXiv: 1804.07461 (cited on page 180).

[676] Wang, C., Riviere, M., Lee, A., Wu, A., Talnikar, C., Haziza, D., Williamson, M., Pino, J., Dupoux, E., "VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation". In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. 2021 (cited on page 202).

[677] Wang, C., Wu, A., Pino, J., *Covost 2 and Massively Multilingual Speech-to-Text Translation*. 2020. arXiv: 2007.10310 (cited on page 203).

[678] Wang, C., Wu, Y., Chen, S., Liu, S., Li, J., Qian, Y., Yang, Z., *Self-Supervised Learning for Speech Recognition with Intermediate Layer Supervision*. 2021. arXiv: 2112.08778. (Visited on 12 April 2023) (cited on page 214).

[679] Wang, C., Wu, Y., Qian, Y., Kumatani, K., Liu, S., Wei, F., Zeng, M., Huang, X., "UniSpeech: Unified Speech Representation Learning with Labeled and Unlabeled Data". 2021. arXiv: 2101.07597 (cited on pages 82, 89, 95).

[680] Wang, G., Rosenberg, A., Chen, Z., Zhang, Y., Ramabhadran, B., Wu, Y., Moreno, P., "Improving Speech Recognition Using Consistent Predictions on Synthesized Speech". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020 (cited on page 220).

[681] Wang, H., Li, Z., Feng, L., Zhang, W., "Vim: Out-of-distribution with Virtual-Logit Matching". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022 (cited on page 27).

[682] Wang, H., Qian, Y., Wang, X., Wang, Y., Wang, C., Liu, S., Yoshioka, T., Li, J., Wang, D., "Improving Noise Robustness of Contrastive Speech Representation Learning with Speech Reconstruction". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022 (cited on page 226).

[683] Wang, Y.-H., Chung, C.-T., Lee, H.-y., "Gate Activation Signal Analysis for Gated Recurrent Neural Networks and Its Correlation with Phoneme Boundaries". In: *Proceedings of the 18th Annual Conference of the International Speech Communication Association (Interspeech)*. ISCA, 2017 (cited on pages 89, 218).

[684] Wang, Y.-H., Lee, H.-y., Lee, L.-s., "Segmental Audio Word2vec: Representing Utterances as Sequences of Vectors with Applications in Spoken Term Detection". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018 (cited on page 218).

[685] Wang, L., Hasegawa-Johnson, M., "A DNN-HMM-DNN Hybrid Model for Discovering Word-like Units from Spoken Captions and Image Regions". In: *Annual Conference of the International Speech Communication Association*. 2020 (cited on page 199).

[686] Wang, L., Luc, P., Wu, Y., Recasens, A., Smaira, L., Brock, A., Jaegle, A., Alayrac, J.-B., Dieleman, S., Carreira, J., Oord, A., *Towards Learning Universal Audio Representations*. 2021. arXiv: 2111.12124 (cited on pages 211, 212).

[687] Wang, S., Thompson, L., Iyyer, M., "Phrase-BERT: Improved Phrase Embeddings from BERT with an Application to Corpus Exploration". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2021 (cited on page 200).

[688] Wang, W., Arora, R., Livescu, K., Bilmes, J. A., "On Deep Multi-View Representation Learning". In: *International Conference on Machine Learning (ICML)*. 2015 (cited on page 198).

[689] Wang, W., Arora, R., Livescu, K., Bilmes, J., "Unsupervised Learning of Acoustic Features via Deep Canonical Correlation Analysis". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015 (cited on page 198).

[690] Wang, W., Tang, Q., Livescu, K., "Unsupervised Pre-training of Bidirectional Speech Encoders via Masked Reconstruction". 2020. arXiv: 2001.10603 (cited on pages 83, 187, 188, 190, 207).

[691]   Wang, W., Yan, X., Lee, H., Livescu, K., *Deep Variational Canonical Correlation Analysis*. 2016. arXiv: 1610.03454. (Visited on 16 March 2023) (cited on page 198).

[692]   Wang, Y., Boumadane, A., Heba, A., *A Fine-Tuned Wav2vec 2.0/HuBERT Benchmark for Speech Emotion Recognition, Speaker Verification and Spoken Language Understanding*. 2021 (cited on page 211).

[693]   Warden, P. *Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition*. 2018. arXiv: 1804.03209 (cited on pages 204, 205).

[694]   Watanabe, S. *Algebraic Geometry and Statistical Learning Theory*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge: Cambridge University Press, 2009 (cited on pages 28, 65).

[695]   Watanabe, S., Opper, M., "Asymptotic Equivalence of Bayes Cross Validation and Widely Applicable Information Criterion in Singular Learning Theory." In: *Journal of Machine Learning Research (JMLR)* 11.12 (2010) (cited on page 28).

[696]   Watkins, C. L., Leathley, M. J., Jones, S. P., Ford, G. A., Quinn, T., Sutton, C. J., "Training Emergency Services' Dispatchers to Recognise Stroke: An Interrupted Time-Series Analysis". In: *BMC Health Services Research* 13 (2013) (cited on pages 9, 140).

[697]   Weinberger, S., Kunath, S., "The Speech Accent Archive: Towards a Typology of English Accents". In: *Language and Computers* 73 (2011) (cited on page 208).

[698]   Wenstrup, J., **Havtorn, J. D.**, Borgholt, L., Blomberg, S. N., Maaloe, L., Sayre, M. R., Christensen, H., Kruuse, C., Christensen, H. C., "A Retrospective Study on Machine Learning-Assisted Stroke Recognition for Medical Helpline Calls". In: *npj Digital Medicine* 6.1 (2023). ISSN: 2398-6352 (cited on pages vii, 13, 139, 166).

[699]   Westbury, J., Milenkovic, P., Weismer, G., Kent, R., "X-Ray Microbeam Speech Production Database". In: *JASA* 88.S1 (1990) (cited on page 198).

[700]   Wildenschild, C., Mehnert, F., Thomsen, R. W., Iversen, H. K., Vestergaard, K., Ingeman, A., Johnsen, S. P., "Registration of Acute Stroke: Validity in the Danish Stroke Registry and the Danish National Registry of Patients". In: *Clinical epidemiology* (2013) (cited on page 149).

[701]   Williams, R. J. "Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning". In: *Journal of Machine Learning* 8.3 (1992) (cited on page 221).

[702]   Wilpon, J., Rabiner, L., "A Modified K-means Clustering Algorithm for Use in Isolated Work Recognition". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 33.3 (1985) (cited on page 178).

[703]   Wilson, D. J. "The Harmonic Mean P-Value for Combining Dependent Tests". In: *Proceedings of the National Academy of Sciences (PNAS)* (2019) (cited on pages 64, 65, 265).

[704]   Winkens, J., Bunel, R., Roy, A. G., Stanforth, R., Natarajan, V., Ledsam, J. R., MacWilliams, P., Kohli, P., Karthikesalingam, A., Kohl, S., "Contrastive Training for Improved Out-of-Distribution Detection". 2020. arXiv: 2007.05566 (cited on page 69).

[705]   Wiskott, L., Sejnowski, T. J., "Slow Feature Analysis: Unsupervised Learning of Invariances". In: *Neural Computation* 14.4 (2002) (cited on pages 82, 186).

[706]   Wolf, T. *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. 2020. arXiv: 1910.03771. (Visited on 31 December 2023) (cited on page 164).

[707]   World Health Organisation (WHO), *International Classification of Diseases (ICD)*. 2023. URL: https://icd.who.int/ (visited on 26 August 2023) (cited on page 10).

[708]   Wrench, A. "A New Resource for Production Modelling in Speech Technology". In: *Procedings of the Institute of Acoustics* 23.3 (2001) (cited on page 198).

[709] Wu, H., Zheng, B., Li, X., Wu, X., Lee, H.-Y., Meng, H., "Characterizing the Adversarial Vulnerability of Speech Self-Supervised Learning". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Singapore: IEEE, 2022 (cited on pages 225, 226).

[710] Wu, Z.-F., Wei, T., Jiang, J., Mao, C., Tang, M., Li, Y.-F., "NGC: A Unified Framework for Learning with Open-World Noisy Data". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021 (cited on page 25).

[711] Xia, P., Wu, S., Van Durme, B., "Which *BERT? A Survey Organizing Contextualized Encoders". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, 2020 (cited on page 178).

[712] Xia, Y., Cao, X., Wen, F., Hua, G., Sun, J., "Learning Discriminative Reconstructions for Unsupervised Outlier Removal". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile, 2015 (cited on pages 24, 29).

[713] Xiao, A., Zheng, W., Keren, G., Le, D., Zhang, F., Fuegen, C., Kalinli, O., Saraf, Y., Mohamed, A., *Scaling ASR Improves Zero and Few Shot Learning*. 2021. arXiv: 2111.05948 (cited on page 181).

[714] Xiao, H., Rasul, K., Vollgraf, R., *Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms*. 2017. arXiv: 2102.06171 (cited on pages 50, 56, 69, 227).

[715] Xiao, Z., Yan, Q., Amit, Y., *Do We Really Need to Learn Representations from In-Domain Data for Outlier Detection?* 2021. arXiv: 2105.09270 (cited on pages 24, 29, 157, 163).

[716] Xiao, Z., Yan, Q., Amit, Y., "Likelihood Regret: An Out-of-Distribution Detection Score for Variational Auto-Encoder". In: *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*. Virtual, 2020 (cited on pages 24, 28, 44, 47, 48, 53, 55, 56, 61, 69).

[717] Xie, X., Xiong, Y., Yu, P. S., Zhu, Y., "EHR Coding with Multi-scale Feature Attention and Structured Knowledge Graph Propagation". In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. CIKM '19. New York, NY, USA: Association for Computing Machinery, 2019. ISBN: 978-1-4503-6976-3 (cited on pages 116, 121, 171).

[718] Xu, Q., Likhomanenko, T., Kahn, J., Hannun, A., Synnaeve, G., Collobert, R., "Iterative Pseudo-Labeling for Speech Recognition". In: *Annual Conference of the International Speech Communication Association*. 2020 (cited on pages 181, 209–211).

[719] Yan, Y., Fung, G., Dy, J. G., Rosales, R., "Medical Coding Classification by Leveraging Inter-Code Relationships". In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '10. New York, NY, USA: Association for Computing Machinery, 2010. ISBN: 978-1-4503-0055-1 (cited on page 125).

[720] Yang, J., Wang, H., Feng, L., Yan, X., Zheng, H., Zhang, W., Liu, Z., "Semantically Coherent Out-of-Distribution Detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021 (cited on page 25).

[721] Yang, J., Zhou, K., Li, Y., Liu, Z., *Generalized Out-of-Distribution Detection: A Survey*. 2022. arXiv: 2110.11334. (Visited on 4 September 2022) (cited on pages 22, 23).

[722] Yang, S.-w., Liu, A. T., Lee, H.-y., "Understanding Self-Attention of Self-Supervised Audio Transformers". In: *Proceedings of the 21st Annual Conference of the International Speech Communication Association (Interspeech)*. ISCA, 2020 (cited on page 214).

[723] Yang, S.-w. "SUPERB: Speech Processing Universal PERformance Benchmark". In: *Annual Conference of the International Speech Communication Association*. 2021 (cited on pages 95, 157, 180, 207–209, 211, 225).

[724] Yang, Z., Wang, S., Rawat, B. P. S., Mitra, A., Yu, H., *Knowledge Injected Prompt Based Fine-tuning for Multi-label Few-shot ICD Coding*. arXiv, 2022. arXiv: 2210.03304 (cited on pages 116, 125).

[725] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q. V., *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. 2019. arXiv: 1906.08237 (cited on page 188).

[726] Yeh, C.-K., Chen, J., Yu, C., Yu, D., "Unsupervised Speech Recognition via Segmental Empirical Output Distribution Matching". In: *Proceedings of the 7th International Conference on Learning Representations (ICLR)*. New Orleans, LA, USA, 2019 (cited on pages 218, 219).

[727] Young, S., Woodland, P., "State Clustering in Hidden Markov Model-Based Continuous Speech Recognition". In: *Computer Speech & Language* 8.4 (1994). ISSN: 0885-2308 (cited on page 179).

[728] Yu, Q., Aizawa, K., "Unsupervised Out-of-Distribution Detection by Maximum Classifier Discrepancy". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019 (cited on page 25).

[729] Yuan, Z., Tan, C., Huang, S., "Code Synonyms Do Matter: Multiple Synonyms Matching Network for Automatic ICD Coding". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Dublin, Ireland: Association for Computational Linguistics, 2022 (cited on pages 120, 121, 136, 171).

[730] Yue, X., Li, H., "Phonetically Motivated Self-Supervised Speech Representation Learning". In: *Annual Conference of the International Speech Communication Association* (2021) (cited on pages 188, 190).

[731] Yun, S., Han, D., Chun, S., Oh, S. J., Yoo, Y., Choe, J., "CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, South Korea: IEEE, 2019 (cited on page 164).

[732] Zadeh, A. B., Liang, P. P., Poria, S., Cambria, E., Morency, L.-P., "Multimodal Language Analysis in the Wild: CMU-MOSEI Dataset and Interpretable Dynamic Fusion Graph". In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. 2018 (cited on page 203).

[733] Zadrozny, B., Elkan, C., "Transforming Classifier Scores into Accurate Multiclass Probability Estimates". In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2002 (cited on pages 11, 165).

[734] Zaeemzadeh, A., Bisagno, N., Sambugaro, Z., Conci, N., Rahnavard, N., Shah, M., "Out-of-Distribution Detection Using Union of 1-Dimensional Subspaces". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021 (cited on pages 24, 27).

[735] Zaken, E. B., Goldberg, Y., Ravfogel, S., "BitFit: Simple Parameter-Efficient Fine-Tuning for Transformer-Based Masked Language-Models". In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. Dublin, Ireland: Association for Computational Linguistics, 2022 (cited on page 224).

[736] Zeiler, M. D., Ranzato, M., Monga, R., Mao, M., Yang, K., Le, Q. V., Nguyen, P., Senior, A., Vanhoucke, V., Dean, J., "On Rectified Linear Units for Speech Processing". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2013 (cited on page 186).

[737] Zhang, L. H., Goldstein, M., Ranganath, R., "Understanding Failures in Out-of-Distribution Detection with Deep Generative Models". In: *Proceedings of the 38th International Conference on Machine Learning (ICML)*. Volume 139. PMLR, 2021 (cited on pages 60, 63, 68).

[738] Zhang, R., Isola, P., Efros, A. A., "Colorful Image Colorization". In: *European Conference on Computer Vision*. 2016 (cited on page 180).

[739] Zhang, Y., Qin, J., Park, D. S., Han, W., Chiu, C.-C., Pang, R., Le, Q. V., Wu, Y., "Pushing the Limits of Semi-Supervised Learning for Automatic Speech Recognition". In: *Proceedings of the Workshop on Self-Supervised Learning for Speech and Audio Processing at NeurIPS*. 2020. arXiv: 2010.10504 (cited on pages 209–211).

[740] Zhang, Z., Liu, J., Razavian, N., "BERT-XML: Large Scale Automated ICD Coding Using BERT Pretraining". In: *Proceedings of the 3rd Clinical Natural Language Processing Workshop*. Online: Association for Computational Linguistics, 2020 (cited on pages 116, 137, 171).

[741] Zhao, S., Song, J., Ermon, S., *InfoVAE: Information Maximizing Variational Autoencoders*. 2018. arXiv: 1706.02262 (cited on page 32).

[742] Zhou, C., Paffenroth, R. C., "Anomaly Detection with Robust Deep Autoencoders". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2017 (cited on pages 24, 29).

[743] Zhou, H., Baevski, A., Auli, M., "A Comparison of Discrete Latent Variable Models for Speech Representation Learning". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021. arXiv: 2010.14230 (cited on page 215).

[744] Zhou, T., Cao, P., Chen, Y., Liu, K., Zhao, J., Niu, K., Chong, W., Liu, S., "Automatic ICD Coding via Interactive Shared Representation Networks with Self-distillation Mechanism". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, 2021 (cited on page 116).

[745] Zhou, X., Liu, H., Pourpanah, F., Zeng, T., Wang, X., "A Survey on Epistemic (Model) Uncertainty in Supervised Learning: Recent Advances and Applications". In: *Neurocomputing* 489 (2022) (cited on page 12).

[746] Zhu, Q.-S., Zhang, J., Zhang, Z.-Q., Wu, M.-H., Fang, X., Dai, L.-R., "A Noise-Robust Self-Supervised Pre-Training Model Based Speech Representation Learning for Automatic Speech Recognition". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022 (cited on page 226).

[747] Zhu, Y., Min, M. R., Kadav, A., Graf, H. P., "S3VAE: Self-supervised Sequential VAE for Representation Disentanglement and Data Generation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA, 2020 (cited on page 105).

[748] Zinger, N. D., Blomberg, S. N., Lippert, F., Krafft, T., Christensen, H. C., "Impact of Integrating Out-of-Hours Services into Emergency Medical Services Copenhagen: A Descriptive Study of Transformational Years". In: *International Journal of Emergency Medicine* 15.1 (2022) (cited on page 148).

[749] Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., Chen, H., "Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. Vancouver, BC, Canada, 2018 (cited on pages 24, 29).