Learned Data Augmentation for Bias Correction

Schwöbel, Pola Elisabeth

Publication date:
2022

Document Version
Publisher's PDF, also known as Version of record

Learned Data Augmentation for Bias Correction

Pola Schwöbel

Kongens Lyngby 2022

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Richard Petersens Plads, building 324,
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Summary (English)

This thesis consists of three independent pieces of research that can be divided into two subject groups. The first block of topics is invariance learning and learned data augmentation (Paper 1 and 2 presented in Chapter 3 and 4, respectively). Paper 1 is concerned with learning invariances (or equivalently, as we will see, data augmentation) via Bayesian model selection and the marginal likelihood. In Paper 2, we take a different approach: achieving invariance by automatically pose-normalising inputs. The second topic block is fairness in machine learning which we cover in Paper 3 (Chapter 6).

In addition to published research, this thesis contains the following original material. The first two chapters introduce the topics and Chapter 5 connects data augmentation with fairness. It investigates whether data augmentation and upsampling can be used make datasets more balanced, and, by correcting data bias, making models more fair. Chapter 7 concludes the work with a summary and discussion.

Summary (Danish)

Denne afhandling består af tre selvstændige forskningsprojekter, der kan opdeles i to grupper. Den første blok af emner er invarianslæring og lært dataforøgelse (artikel 1 og 2 præsenteret i henholdsvis kapitel 3 og 4). Det første arbejde i denne blok handler om læring af invarianser (eller tilsvarende, som vi vil se, data augmentation) via Bayesiansk modeludvælgelse og den marginale sandsynlighed. I den anden artikel tager vi en anden tilgang: at opnå invarians ved automatisk pose-normalisering af input. Det andet emne, der behandles, er fairness i maskinlæring (artikel 3, kapitel 6).

Ud over publiceret forskning indeholder denne afhandling følgende originale materiale. De første to kapitler introducerer emnerne, og kapitel 5 forbinder dataforøgelse med ML fairness. Det undersøger, om dataforøgelse og upsampling kan bruges til at gøre datasæt mere afbalancerede og ved at korrigere databias gøre modeller mere fair. Kapitel 7 afslutter arbejdet med en opsummering og diskussion.

Preface

This thesis was written at the Section for Cognitive Systems, DTU Compute, Technical University of Denmark in fulfillment of the requirements for acquiring a PhD degree at the Technical University of Denmark. Professor Søren Hauberg and associate professor Kristoffer Hougaard Madsen supervised the project. The project was funded by a DTU Compute scholarship.

The thesis work was carried out from January 2019 to June 2022 at the Technical University of Denmark, with an exception of four months external stay at Imperial College London (remotely in part due to the COVID-19 pandemic). The supervision at this time was conducted by Mark van der Wilk. The project work was also paused for four months in 2021 for an internship at Amazon under the supervision of James Hensman in Cambridge, UK.

The work of this thesis amounts to three papers, and is presented with a thorough introduction and well as theoretical and experimental work connecting the papers. All papers are appended to this thesis.

Lyngby, 07-06-2022

Pola Schwöbel

Acknowledgements

Doing a PhD at the Section for Cognitive Systems has been wonderful. I have never before learned so much in so little time, and I am thankful to many people for sharing and shaping the experience.

Thank you, first of all, to my advisor Søren Hauberg who taught me how to be a better scientist and a better human, to build intuitions and to follow them. Thanks to my co-advisor Kristoffer H. Madsen for bringing in new perspectives. Thanks to Mark van der Wilk for advising me during my research stay at Imperial College London, and for teaching me how to work from first principles. Thanks to James Hensman for advising me during my internship at Amazon.

I am incredibly thankful to my brilliant collaborators and colleagues who have made this experience fun, and who have taught me many things big and small. To mention but a few: thanks Martin for teaching me lots about statistics, LaTeX and keeping one's cool under pressure. Thanks Frederik and Nicki for teaching me how to write beautiful code, and to Cilie for many inspiring discussions about all things feminism and Gaussian processes. Thanks to Ejner Fergo for helping all of us using compute infrastructure with infinite patience.

I thank my family, especially my sister, my father and Oma Anneliese, for everything. And, of course, my mother, who I am missing on this day and everyday. Thanks to my friends for putting up with me during the process. Lastly, thanks to Sebastian for teaching me how to share a life with somebody, the most exciting learning of all.

List of Publications

Published

Last Layer Marginal Likelihood for Invariance Learning
Pola Schwöbel, Martin Jørgensen, Sebastian W. Ober and Mark Van Der Wilk
25th International Conference on Artificial Intelligence and Statistics (AISTATS), 2022
(Paper 1)

Probabilistic Spatial Transformer Networks
Pola Schwöbel, Frederik Warburg, Kristoffer H. Madsen and Søren Hauberg
38th Conference on Uncertainty in Artificial Intelligence (UAI), 2022
(Paper 2)

The Long Arc of Fairness: Formalisations and Ethical Discourse
Pola Schwöbel and Peter Remmers (equal contribution)
ACM Conference on Fairness, Accountability, and Transparency (ACM FAccT), 2022
(Paper 3)

Unpublished Material

The case study in Chapter 5 contains original material which was produced for this thesis.

Contents

Chapter 1

Introduction

Machine learning algorithms work by extracting patterns from large amounts of training data, and projecting existing correlations forward. This mechanism by which ML works becomes a problem when training data is scarce: In the low data regime, performance usually drops dramatically. Dangerously, training data can be lacking in *systematic ways*.

For example, in the medical domain, convolutional neural networks are reported to detect melanoma (a dangerous type of skin cancer) very successfully, at rates outperforming human dermatologists [Brinker et al., 2019]. As Norori et al. [2021] point out performance outside a controlled lab setting, on a more diverse population, looks very differently: Skin disease classifiers are trained on a predominantly white population[1] and, as a consequence, their accuracy drops to about half of what was originally reported when evaluated on a majority black population [Kamulegeya et al., 2019]. This results in under- and misdiagnosis of Black patients who are already suffering higher melanoma mortality rates [Norori et al., 2021].

In fact, when we talk about algorithmic bias in the context of ML, this bias can often be traced back to such *data* bias. Buolamwini and Gebru [2018]

[1] Kamulegeya et al. [2019] estimate 5-10% Black patients among the training data used for First Derm's Skin Image Search software, an ML-based skin disease classifier (`https://www.firstderm.com/ai-dermatology`.

show how commercial facial recognition software routinely fails on women of color, because they are underrepresented in the datasets these algorithms are trained on. A sexist hiring algorithm prefers male candidates, presumably because it is trained on a predominantly male tech work force [Dastin, 2018]. Search engines perpetuate stereotypes about women of color by surfacing media reflecting historic racism [Noble, 2018].

As a consequence, one might look to the *data* to alleviate algorithmic biases. While (re-)collecting unbiased datasets would be the measure of choice, this might be unfeasible for economic or practical reasons, or fundamentally impossible (e.g. when studying rare diseases the amount of examples, especially amongst a demographic minority, is limited by nature). Thus, this work aims to investigate whether synthetic data might help in low and biased data regimes.

The idea is to use *data augmentation* (DA), an established ML engineering practice to generate new data by making small modifications to existing data. For example, a new image can be generated by slightly rotating an existing one. We investigate whether we can use DA to upsample parts of the dataset in order to combat bias.

Chapter 2 introduces standard data augmentation. DA is usually performed *ad hoc*, i.e. based on assumptions rather than using a statistical model. However, ad hoc DA following no principled model cannot guarantee optimal augmentation, and might be difficult to do for data that is not easily interpreted by humans. Thus, the second chapter deals with formalising the classic ad hoc practice and providing a principled model for data augmentation. We will close this chapter by giving a theoretical argument for why *learning* data augmentation is more difficult than one might expect.

Chapter 3 will then present a way to learn data augmentation (or, equivalently as we will see, invariances) nonetheless: Rather than learning it using the usual maximum-likelihood loss, we rephrase the problem as Bayesian model selection and learn invariances by maximising the marginal likelihood. In order to be able to compute this quantity, we utilise deep kernel GPs; hybrid models combining elements of neural networks with Gaussian processes.

Chapter 4 investigates another way to achieve model invariance. Instead of augmenting the data, i.e. showing data to our model in all relevant poses, we automatically learn to pose-normalise the data. This idea is implemented via a probabilistic extension to spatial transformer networks.

With these data augmentation and invariance learning strategies, we have developed the necessary tools to test our hypothesis: Can data augmentation

be used to alleviate data bias and, as a consequence, produce fairer models? Chapter 5 contains a case study investigating this experimentally.

In the case study we aim to 'automatically' de-bias an ML algorithm with data augmentation. To evaluate whether we are successful, we depend on quantitative, i.e. formalized criteria for unbiasedness. When reviewing the large body of literature on such fairness metrics it becomes increasingly clear that none of them can be considered 'the single right metric' to optimize for. What's 'fair' depends crucially on context and, of course, on moral commitments; an aspect that has seen relatively little analysis in the fairness literature. Thus, Chapter 6 takes a broader view on ethics and fairness modelling. Building on Paper 3 [Schwöbel and Remmers, 2022] we introduce a modelling framework which is more contextualized than most existing approaches, hereby hoping to bridge ethical and formalized debates and embedding the work of into a larger context.

Chapter 7 concludes this thesis with a summary and discussion. Appended are the three papers this work is built on.

Chapter 2

Data Augmentation and Invariance

In supervised machine learning, we aim to fit a function f to $N_{\mathcal{D}}$ labelled data points $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N_{\mathcal{D}}}$. In the parametric case f, is fully described by its parameters w, e.g. the weights of a neural network. Fitting the model then corresponds to finding w such that

$$f_w(x_i) \approx y_i \text{ for all } i = 1 \dots N_{\mathcal{D}}. \tag{2.1}$$

We will discuss both such deterministic models as well as probabilistic formulations in the following chapter, and we will switch between the two whenever it helps to build intuitions. In a probabilistic model parameterised by the same weights w, we aim to, equivalently, maximise the probability assigned to the correct label,

$$p_w(y_i|x_i) \approx 1, \text{ for all } i = 1 \dots N_{\mathcal{D}}. \tag{2.2}$$

The problem in Eq. 2.1 is well-posed whenever the number of data points $N_{\mathcal{D}}$ is at least equal to the number of parameters N_w. However, this is usually not the case for modern neural network models which often consist of millions or billions of parameters but are routinely trained on datasets which are orders of magnitude smaller in size. As a consequence, one can increase the dataset size by adding synthetic data (i.e. applying data augmentation), reducing the

expressivity and thus effective size of the model by imposing inductive biases (e.g. in convolutional neural networks), or applying other regularisation techniques.

When $N_w \gg N_{\mathcal{D}}$ Equation 2.1 has many solutions. Bad solutions are characterised by overfitting, i.e. they minimise the loss on training data but perform worse on test data.

We pause to state some assumptions about the data that we will make, and that will turn out to be important in the context of data augmentation. We assume all data to be i.i.d.. In particular, training and test data follow the same distribution

$$\mathcal{D} \sim p(\mathcal{D}) = p(\mathcal{D}_{test}) \sim \mathcal{D}_{test} \tag{2.3}$$

and datapoints are independent, i.e.

$$p(\mathcal{D}) = \prod_{i=1}^{N_{\mathcal{D}}} p(x_i, y_i). \tag{2.4}$$

The independence assumption (2.4) is often violated in standard data augmentation, we will discuss this in Sec. 2.2.1.

2.1 Standard Data Augmentation

If we had access to the true data generating process $p(\mathcal{D})$, we could easily alleviate the overfitting problem, since we could draw arbitrarily many new samples (x_i, y_i), hereby increasing $N_{\mathcal{D}}$. The problem would be less ill-posed (since we could make $N_{\mathcal{D}}$ be as large as N_w). In other words, we could make it arbitrarily unlikely to sample a new test point that is far away from training data. This approach is mimicked in standard data augmentation work, where one aims to 'close the gaps' in $\mathcal{D}$ by making assumptions about the data generating process.

DA is particularly common for image data, where such assumptions can be made straight-forwardly. One often assumes that a new image can be generated from an existing one via an affine transformation, i.e. by rotating, scaling or translating in early work by LeCun et al. [1995] and Loosli et al. [2007]. Simard et al. [2003] additionally consider elastic distortions. Prominently, Krizhevsky et al. [2012] mention data augmentation in their seminal work on deep neural networks as one of the factors allowing them to train deep models in the first

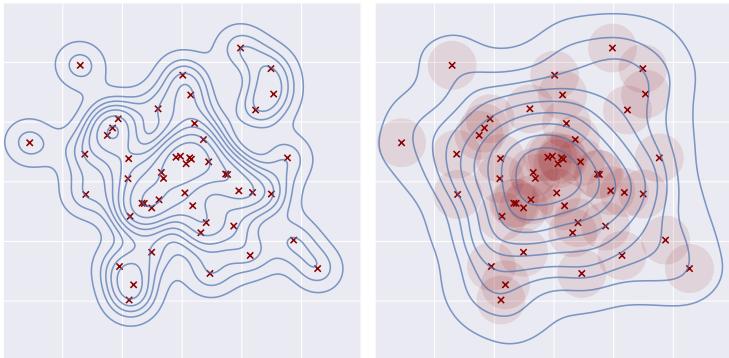


Figure 2.1: *Left:* A model of $p(D)$ (blue) after observing 50 samples (red). *Right:* A model fitted on 50 samples plus data augmentation. The data augmentation scheme is to add Gaussian noise to each point (i.e. sample roughly within the red circles). The model of $p(D)$ fitted on augmented data is smoother. As we will see in Sec. 2.2.2.1, data augmentation is equivalent to regularising the model to be smooth.

place. Without DA, they explain, they would have been 'forced to use much smaller networks'. They use random translations, reflections as well as color space augmentations (along the principal components of the 3D color space). Today, data augmentation is a standard practice for training deep models in the vision domain. Torchvision's data augmentation toolbox[1], for example, provides more than 20 image augmentation strategies that can be applied out of the box.

2.2 Beyond Standard Data Augmentation

While being an extremely useful engineering trick, standard DA schemes suffer from a range of shortcomings. We will discuss two such shortcomings here and offer solutions from the literature as well as our own research.

[1] `https://pytorch.org/vision/0.12/transforms.html`

2.2.1 A Principled Model

As discussed above, data augmentation usually creates new images from existing ones by applying transformations. The new, augmented dataset is

$$\widetilde{\mathcal{D}} = \{(T^j(x_i), y_i)\}_{i,j},\ i = 1, ..., N_{\mathcal{D}},\ j = 1, ...N_T \tag{2.5}$$

for transformations T^j. N_T is the number of augmentations and is typically equal to the number of epochs the model is trained for, i.e. a new transformation is sampled each time the model sees the datapoint (x_i, y_i). From this formulation, it becomes obvious that the independence assumption from Eq. 2.4 is violated for standard data augmentation: we create a new data point *from an existing one.*

Taking a probabilistic viewpoint, a solution becomes available. We may think of data augmentation as *marginalising transformations*

$$p_w(y|x) = \int p_w(y, T|x)\ \mathrm{d}T = \int p_{w_p}(y|T, x) p_{w_a}(T|x)\ \mathrm{d}T, \tag{2.6}$$

hereby appropriately capturing the relation between transformed images. In the above equation we divide the set of weights into weights parametrising the predictor w_p and the augmentation distribution w_a, such that $w_p \cup w_a = w$.

Early works taking this approach [Chapelle et al., 2000, Maaten et al., 2013] use simple augmentation distributions and simple model classes (e.g. p_{w_a} is a Gaussian distribution over point-wise noise and p_w is a linear model), such that the integral (2.6) can be computed in closed form, or approximated easily. Recent works such as Benton et al. [2020], van der Wilk et al. [2018], Schwöbel et al. [2021, 2020] use the same modelling assumptions, we will discuss these later (Sec. 2.2.2.2, Ch. 3 and Ch. 4).

This principled model correctly captures the covariance structure between augmented images but it, of course, crucially depends on the augmentation distribution $p_{w_a}(T|x)$. As we have seen in Sec. 2.1, standard approaches rely on assumptions, for example that a reasonable augmentation distribution might be rotations by $\pm 15°$. While it is relatively easy to make approximately correct assumptions for many types of natural images, one can easily imagine cases where this is not straight forward. As a consequence, recent research efforts aim to *learn* data augmentation, i.e. to infer a suitable $p_{w_a}(T|x)$.

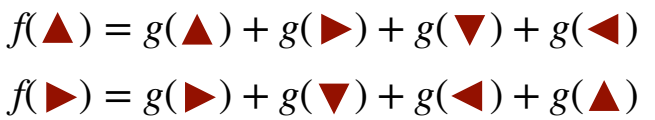


Figure 2.2: Visual representation of the invariance construction from Eq. 2.6 with a discrete transformation distribution (the four rotations as described in Sec. 2.2.1.1). The RHS is the same for the original input in the first row and the rotated input in the second row, i.e. f is invariant with respect to 90°-rotations.

2.2.1.1 Data Augmentation and Invariance

Before discussing approaches to learning the transformation distribution $p_{w_a}(T|x)$, we pause to comment on terminology. Works such as Benton et al. [2020], van der Wilk et al. [2018] and Schwöbel et al. [2021] refer to estimating $p_{w_a}(T|x)$ as learning *invariances* rather than learning *data augmentation*.

Definition 2.1 (Invariance) A function f is invariant w.r.t. transformation T iff

$$f(x) = f(T(x)) \tag{2.7}$$

for all x.

Now, consider the construction in Eq. 2.6 for some finite $p_{w_a}(T|x)$. For example, if x is an image and T are rotations by 90°, we arrive at the original x after applying T four times. The group G of 90° rotations has 4 elements, $G = \{T^1 := T_{90^\circ}, T^2 := T_{180^\circ}, T^3 := T_{270^\circ}, T^4 := T_{360^\circ} = T_{0^\circ} = \mathrm{I}\}$. We might sample each of the 4 rotations with equal probability. Then, if we define f similar to Eq. 2.6, by marginalising transformations over a non-invariant function g, we have

$$f(x) = \int f(T(x))p(T|x)\mathrm{d}T = \frac{1}{4}\sum_{i=1}^{4} g(T^i(x)) \text{ and} \tag{2.8}$$

$$f(T^1(x)) = \frac{1}{4}\sum_{i=1}^{4} g(T^i(T^1(x))) = \frac{1}{4}\sum_{i=1}^{4} g(T^{i+1}(x))) \tag{2.9}$$

$$= \frac{1}{4}\sum_{i=2}^{4} g(T^i(x))) + g(T^1(x)) = f(x). \tag{2.10}$$

Hence $f(x) = f(T^1(x))$, i.e. the construction in Eq. 2.6 yields an invariant function with respect to 90°-rotations. See Fig. 2.2 for an illustration.

Another way to look at the connection between data augmentation and model invariance is the following: Recall that when applying data augmentation, one usually only augments inputs $T(x)$ while leaving labels y unaltered (see Eq. 2.5). Thus, when training a model f to perfectly fit the datapoints (x_i, y_i) and $(T(x_i), y_i)$ then $y_i = f(x_i) = f(T(x_i))$, i.e. the function is invariant to T according to Def. 2.1.

When talking about data augmentation we usually imagine T to refer to small perturbations, whereas literature on invariances usually considers more drastic transformations T. This difference in naming is purely conventional and we will use the terms 'learned data augmentation' and 'invariance learning' interchangeably.

2.2.2 Learned Data Augmentation

Naively, one might try to learn the optimal data augmentation strategy (equivalently, the appropriate model invariances) in the same way one learns the model weights, i.e. by finding w such that $f_w(x_i) \approx y_i$ for all $i = 1, \dots N_{\mathcal{D}}$ by minimising a loss function. We will now show that for standard loss functions, such as the mean squared error (MSE) loss, this will result in *collapsing augmentation distributions*, i.e. learning that we should not augment at all.

2.2.2.1 The Naive Approach and Why It Fails

Consider a simple $1d$ regression case, i.e. $f_w : \mathbb{R} \to \mathbb{R}$. Assume that the augmentation distribution is additive Gaussian pixel noise $T(x) = x + \varepsilon$ with $p_{w_a}(T) = \mathcal{N}(\varepsilon | 0, w_a^2 \operatorname{Id})$, i.e. w_a is a scalar variance parameter. We are fitting our model by minimising the mean squared error

$$E = \iint (f_w(x) - y)^2 p(x, y) \, \mathrm{d}x \, \mathrm{d}y. \tag{2.11}$$

Let $\widetilde{E}$ denote the MSE loss after applying data augmentation, i.e.

$$\widetilde{E} = \iiint (f_w(x + \varepsilon) - y)^2 p(x, y) p(\varepsilon) \, \mathrm{d}x \, \mathrm{d}y \, \mathrm{d}\varepsilon. \tag{2.12}$$

We start by Taylor-expanding f_w in x, omitting w for brevity:

$$f(x + \varepsilon) = f(x) + \varepsilon f'(x) + \frac{1}{2} \varepsilon^2 f''(x) + \mathcal{O}(\varepsilon^3). \tag{2.13}$$

Plugging this back into Eq. 2.12 and omitting the cubic error terms (these are small given that ε corresponds to small perturbations of the input), we obtain

$$\widetilde{E} \approx \iiint (f(x) - y + \varepsilon f'(x) + \frac{1}{2}\varepsilon^2 f''(x))^2 p(x,y)p(\varepsilon)\ \mathrm{d}x\ \mathrm{d}y\ \mathrm{d}\varepsilon \tag{2.14}$$

$$= \iiint (f(x) - y)^2 + 2(f(x) - y)(\varepsilon f'(x) + \frac{1}{2}\varepsilon^2 f''(x)) + (\varepsilon f'(x) + \frac{1}{2}\varepsilon^2 f''(x))^2 p(x,y)p(\varepsilon)\ \mathrm{d}x\ \mathrm{d}y\ \mathrm{d}\varepsilon \tag{2.15}$$

$$= E + w_a^2 \iint f'(x)^2 + (f(x) - y)f''(x)p(x,y)\ \mathrm{d}x\ \mathrm{d}y \tag{2.16}$$

$$=: E + w_a^2 \Omega(f). \tag{2.17}$$

From Eq. 2.15 to Eq. 2.16 we have evaluated the integral with respect to ε, $\int \varepsilon\ d\varepsilon = 0$ and $\int \varepsilon^2\ d\varepsilon = w_a^2$. We have also omitted higher order terms in ε, these are absorbed in $\mathcal{O}(\varepsilon^3)$.

As shown in Bishop [1995] E is minimised by $f^{\min}(x) = \mathbb{E}[y|x]$. He then argues that $\widetilde{E}$ is minimised by $\tilde{f}^{\min}(x) = \mathbb{E}[y|x] + \mathcal{O}(w_a^2)$. Plugging $\tilde{f}^{\min}(x)$ into Eq. 2.15 we see that, in expectation, $\tilde{f}^{\min}(x) - y \approx 0$ (up to $\mathcal{O}(w_a^2)$), and hence

$$\Omega(f) \approx \int f'(x)^2\ \mathrm{d}x. \tag{2.18}$$

Thus, $\Omega(f)$ has the standard form of a Tikhonov regulariser. In particular, the regularisation term is positive, and thus $\widetilde{E}$ will be minimal if $w_a = 0$. As a consequence, aiming to learn w_a by optimising the MSE (2.12) will yield a collapsing augmentation distribution. For an experimental illustration see Fig. 2.3.

Bishop [1995] derives this result for higher dimensions, and for a more general case in Bishop and Nasrabadi [2006], Ch. 5.5.5: For a general augmentation distribution $p(T)$, i.e. $T(x)$ is an arbitrary transformation, we can obtain a similar result by applying a Taylor expansion to $T(x)$ as well. The regulariser term $\Omega(f)$ is then equivalent to the tangent prop regulariser [Simard et al., 1992], encouraging f_w to be constant along the tangent direction of the manifold spanned by $T(x)$.

2.2.2.2 Data Augmentation as Regularisation

In the previous section we have established that data augmentation can be thought of as regularisation. Learning the magnitude of the augmentation as

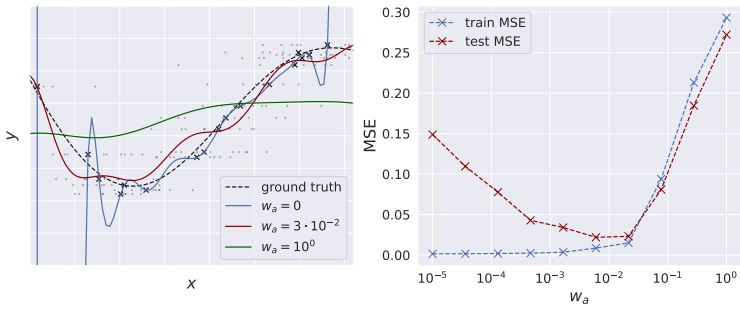


Figure 2.3: *Left:* Different models fitted to training data (black x's). The model with no augmentation (blue) overfits, while the model with too much augmentation underfits (green). The red model is fit with a suitable amount of data augmentation, the samples from this optimal augmentation distribution are plotted in gray. Note how these are horizontal lines along existing data points as we only augment inputs x. *Right:* Train (blue) and test (red) MSE as a function of the magnitude of augmentation applied. As suggested by theory, train MSE increases with increasing w_a. Test MSE, however, decreases at first, i.e. while DA does not improve the fit on training data it helps us generalise.

encoded by w_a corresponds to learning how much to regularise. However, regularisation does not help to improve the fit on training data. In fact, when we regularise we trade a worse fit on training data for better generalisation. Consequently, as we've seen in Sec. 2.2.2.1, $w_a = 0$ minimises MSE (or maximises log-likelihood), and so these standard losses cannot be used to learn w_a. How then can we *learn* data augmentation?

Given that DA helps generalisation, one approach is to use held-out data to determine the optimal augmentation strategy. On the validation set one can then optimise the augmentation parameters using different strategies. A brute force grid or random search might be sufficient if the parameter space is low-dimensional [Cubuk et al., 2020]. If the space is bigger but the augmentations are differentiable, it is possible to apply (meta-)gradient descent on the validation loss [Lorraine et al., 2020]. For non-differentiable transformation distributions, one can resort to reinforcement learning [Cubuk et al., 2019]. Benton et al. [2020] argue that in their application, the loss function is relatively flat w.r.t. w_a, and so they can avoid costly cross-validation to determine the optimal parameter (since any 'small' w_a works). We do not find this to be the case in our somewhat similar setup in Chapter 4, where we depend on fine-tuning the magnitude of

augmentation. Another approach is to do away with end-to-end training all together and train an unsupervised data augmentation model in a separate pre-training step: Hauberg et al. [2016] align images pairwise and fit a per-class distribution $p(T)$ which they can sample from to obtain augmentation transformations.

While these methods are data-driven, they do not *learn* data augmentation end-to-end in the conventional sense of minimising a loss function on training data. Indeed, this is impossible for the standard loss functions MSE and negative log-likelihood as we have seen in Sec. 2.2.2.1. In the next chapter, we will discuss an alternative approach.

Chapter 3

Invariance Learning via Bayesian Model Selection

In the last chapter we have phrased learning data augmentation as determining the optimal amount of regularisation, and thus, in a sense, the right model complexity. Under this viewpoint a new way of thinking about learned data augmentation becomes available: We can think of picking the right augmentation distribution as a *model selection* problem. From a Bayesian perspective, we can perform model selection via type II max-likelihood, i.e. by using the *marginal likelihood* — this perspective on invariance learning is introduced by van der Wilk et al. [2018].

3.1 Marginal Likelihood for Model Selection

Fitting a model implies finding the right parameters using max-likelihood (or related loss functions like MSE), i.e. finding the optimal w_p for a given model class f_{w_p}. For example, in Fig. 2.3, we fit a degree 15-polynomial by estimating its 16 coefficients. Model selection, on the other hand, refers to the problem of finding the right model class. Instead of determining the 16 optimal coefficients, goal here is to determine *which degree* of polynomial to choose in the first place. To give another example, training a neural network corresponds to finding the

optimal weights, whereas model selection requires finding the right architecture (e.g. width and depth, inductive biases such as convolutions, and so on).

In Bayesian inference we perform this task by maximising the marginal likelihood

$$p(Y|X,\theta) = \int p(Y|X,w_p)p(w_p|\theta)\ \mathrm{d}w_p. \tag{3.1}$$

with respect to θ. Here, $X = \{x_1, \ldots, x_{N_\mathcal{D}}\}$ is collecting all the input data, and similarly $Y = \{y_1, \ldots, y_{N_\mathcal{D}}\}$ contains all labels. θ corresponds to the model hyperparameters (i.e. θ encodes the fact that we use a 16th-degree polynomial in the first example, or the number and width of layers in the neural network example). Note that, in the usual Bayesian framework, we are often concerned with a third 'layer' of inference, the class of models (or hypotheses $\mathcal{H}$). For example, $\mathcal{H}_1$ might refer to the model class of all possible neural networks, $\mathcal{H}_2$ to the model class of polynomials, and so on. Performing marginal likelihood optimisation w.r.t. $\mathcal{H}$ is often unfeasible as it would require computations such as integrating over all possible neural network architectures. Thus, in practice, we often decide on the hypothesis class $\mathcal{H}$ in advance and optimise the respective model hyperparameters (see e.g. Williams and Rasmussen [2006], Ch. 5).

The marginal likelihood as a loss function for model selection provides an automatic way to pick the right model complexity (what some authors refer to as an automatic Occam's razor [Williams and Rasmussen, 2006, Rasmussen and Ghahramani, 2001, Lotfi et al., 2022]). To understand why this is the case, consider the schematic illustration in Fig. 3.1, left: The marginal likelihood $p(Y|X,\theta)$ is a probability distribution, i.e. for any given θ it integrates to 1. Very flexible models with many parameters (θ^3, green) can explain a wide range of datasets, but assign less mass to individual datasets because their density must integrate to 1. Conversely, simple models (θ^1, blue) concentrate their mass around fewer datasets and, as a consequence, explain those better. For this reason, the marginal likelihood can find models with the optimal complexity (θ^2, red), not too complex to 'spread their mass too thinly' across many datasets, but complex enough to explain the dataset at hand Y_0.

To apply this mechanism to learning data augmentation, we have to make the the following two modifications to our setup. Firstly, we have to view the data augmentation parameters w_a as model hyperparameters, i.e. $w_a \in \theta$. This is a reasonable modelling assumption given the 'data augmentation as regularisation' view that we have developed in Sec. 2.2.2: The chosen regulariser will determine which model we learn, i.e. $w_a \to f_{w_p}$ in a graphical model (Fig. 3.1, right). Secondly, we have to marginalise over the model parameters w_p to obtain the marginal likelihood from Eq. 3.1.

Using the marginal likelihood loss function instead of regular maximum-likelihood, we can optimise for the data augmentation parameters w_a directly. However, investigating Eq. 3.1 it becomes clear that this quantity is difficult to compute for parameter-rich models f_{w_p} such as neural networks. On the other hand, for models with tractable marginal likelihoods such as Gaussian processes, it can be used to successfully learn invariances [Van der Wilk et al., 2018]. We will shortly review their work (Sec. 3.2.4) and our extension thereof (Sec. 3.3) in detail. Before doing so we pause here for some background material.

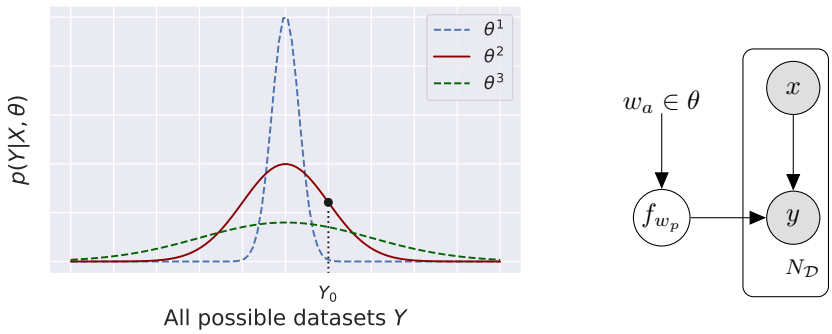


Figure 3.1: *Left:* Visualisation of model selection via the marginal likelihood, adapted from Williams and Rasmussen [2006], Chapter 5. *Right:* A graphical representation of the model structure that we assume in order to perform invariance learning via Bayesian model selection, i.e. by using the marginal likelihood. Grey nodes are observables and white are latents.

3.2 Background

In the last section, we have established that the marginal likelihood is a promising loss function for model selection and hence invariance learning. Recall that the marginal likelihood is computed by marginalising over *all possible functions* (3.1). In the case of neural networks, this would require us to compute an integral over all possible weights of a neural network, i.e. $p(w_p|\theta)$ could be a distribution over many millions of parameters. We face intractability issues.

3.2.1 Bayesian Deep Learning

Bayesian deep learning (BDL) aims to estimate posteriors over neural network weights, and approximate integrals such as Eq. 3.1. By reasoning about weight distributions rather than point estimates, authors achieve improved uncertainty quantification, robustness and predictive performance. For computational reasons, the chosen approximations are often rather crude. For example, Blundell et al. [2015] choose a mean-field approximation where all weights are modelled to follow independent Gaussian distributions. Other works [MacKay, 1992, Daxberger et al., 2021] also make Gaussians assumptions, but use a Laplace approximation to the posterior instead of variational techniques. Gal and Ghahramani [2016] propose to interpret dropout as a Bernoulli approximation to the weight posterior (i.e. they obtain samples from a posterior over networks by randomly switching each weight on or off). A conceptually simple yet very successful approach for BDL is to train an ensemble of methods and interpret the different, trained models as samples from a weight posterior [Lakshminarayanan et al., 2017, Gustafsson et al., 2020]. Generally, such approximate weight posteriors are useful in practice for performance, robustness and uncertainty quantification, the marginal likelihood estimates are typically too imprecise for hyperparameter estimation [Blundell et al., 2015, Turner and Sahani, 2011]. Thus, instead of relying on such approximate methods, we will use Gaussian process models.

3.2.2 Gaussian Processes

Gaussian processes are a class of models for which, unlike neural networks, the marginal likelihood is available in closed form. We review them here.

Definition 3.1 (Gaussian Process) A Gaussian process (GP) [Williams and Rasmussen, 2006] is a distribution over functions $f : \mathcal{X} \to \mathbb{R}$ such that any vector of function evaluations $(f(x_1), \ldots, f(x_N))^\intercal$ follows a Gaussian distribution. We write

$$f(x) \sim \mathcal{GP}(\mu(x), k(x, x')) \tag{3.2}$$

where $\mu(x) = \mathbb{E}[f(x)]$ is the mean and $k(x, x') = \mathbb{E}[(f(x) - \mu(x))(f(x') - \mu(x'))]$ is the covariance function of the process.

We usually assume zero prior mean functions $\mu(x) \equiv 0$ and real valued vector inputs, i.e. $x \in \mathcal{X} = \mathbb{R}^d$. For illustrative purposes we consider a one-dimensional output domain, but the derivations extend straight-forwardly

to multiple independent output dimensions (e.g. in the 10-class classification examples we will see later, $f : \mathcal{X} \to \mathbb{R}^{10}$, the logits of the class probabilities).

Any covariance function that yields a positive semi-definite kernel matrix and is differentiable with respect to its hyperparameters can be used, and, since choosing a kernel corresponds to specifying a prior, the choice of kernel should ideally be made based on domain knowledge about the data. In practice the radial basis (or squared exponential) covariance function is a popular choice,

$$k(x, x') = s^2 \exp\left(-\frac{\|x - x'\|^2}{2\ell^2}\right), \tag{3.3}$$

where s^2 is the kernel variance and ℓ the kernel lengthscale. k will refer to this function in the remainder of this chapter.

We note that GPs are *non-parametric* models, i.e. they are fully described by their mean and covariance functions and the data itself. Instead of reasoning about model parameters w_p in the neural network setting, GPs perform inference over function objects directly. We will thus switch notation here from denoting the target function by its parameters w_p (e.g. Eq. 3.1) to the function object f (e.g. Eq. 3.2) itself.

When f is a Gaussian process as defined in Eq. 3.1 and $p(y|f(x)) = \mathcal{N}(y|f(x), \sigma^2)$ is a Gaussian likelihood, the marginal likelihood in Eq. 3.1 as well as the posterior predictive can, in principle, be computed in closed form thanks to the analytical properties of Gaussian distributions. We can make predictions at a new test point x by conditioning on the already seen data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N_\mathcal{D}}$ collected in $\{X, Y\}$ via

$$f(x)|\mathcal{D} \sim \mathcal{N}(\widetilde{\mu}(x), \widetilde{K}) \tag{3.4}$$

$$\text{with } \widetilde{\mu}(x) = k(x, X)[k(X, X) + \sigma^2 I]^{-1} Y \tag{3.5}$$

$$\text{and } \widetilde{K} = k(x, x) - k(x, X)[k(X, X) + \sigma^2 I]^{-1} k(X, x). \tag{3.6}$$

Despite the existence of closed-form solutions, evaluating these expressions is difficult as Eq. 3.4 and 3.5 involve inverting the $N_\mathcal{D} \times N_\mathcal{D}$ matrix $[k(X, X) + \sigma^2] = [k(x_i, x_j)]_{i,j=1}^{N_\mathcal{D}} + \sigma^2 I$. This is computationally prohibitive even for medium-sized datasets.

To overcome this problem and make GPs scalable one can make use of sparse approximations, for example the sparse variational approximation by Hensman et al. [2015]. We consider $N_Z \ll N_\mathcal{D}$ auxiliary datapoints that in some sense, represent the true data well and give rise to feasible computations. For this approximation, let $Z = \{z_i\}_{i=1}^{N_Z} \in \mathcal{X}$ be the inducing inputs, $U = \{u_i\}_{i=1}^{N_Z} =$

$f(Z) \sim \mathcal{N}(m, S)$ the function values at those inputs. The outputs again follow a Gaussian distribution with m being the variational mean and S the variational covariance matrix at the inputs Z. Z, m and S are optimised freely. This gives rise to a variational posterior

$$q(f) = \mathcal{GP}(\mu(\cdot), \nu(\cdot, \cdot)) \tag{3.7}$$

and the corresponding ELBO

$$\log p(Y) \leq \mathcal{L} = \sum_{i=1}^{N_{\mathcal{D}}} \mathbb{E}_{q(f(x_i))}[\log p(y_i|f(x_i))] - \mathrm{KL}(q(U)||p(U)). \tag{3.8}$$

This formulation allows for mini-batching and thus scaling to large datasets, as well as for non-Gaussian likelihoods in which case we can evaluate the variational expectation $\mathbb{E}_{q(f(x_i))}[\log p(y_i|f(x_i))]$ via Monte Carlo sampling. For these reasons, we mainly work with this approximation in Paper 1 (Sec. 3.3). We note, however, that this choice of variational approximation implies a large amount of free parameters (in Z, m and S) and can hence be difficult to optimise. Thus, we resort to a second approximation in Paper 1, Sec. 7: For Gaussian likelihoods, we can compute the optimal m and S analytically rather than optimising them using gradients [Titsias, 2009]. The resulting approximation has fewer variational parameters (we optimise the inducing inputs Z only) and is easier to optimise. On top of the need for Gaussian likelihoods this approximation also does not allow for mini-batching, hence we need to pre-exact lower dimensional features in this experiment. For details, see Paper 1, Sec. 7 or its summary in Sec 3.3.3 (CIFAR-10 experiment).

3.2.3 Differentiable Transformation Distributions

In order to learn data augmentation we want to backpropagate through the distribution of transformations, and thus our transformations T need to be differentiable. A typical and simple class of transformations are affine transformations T_ϕ with $\phi = (\alpha, s^x, s^y, p^x, p^y, t^x, t^y)$, where the parameters describe rotation, scale, shearing and horizontal and vertical translation. A transformation T_ϕ is applied to image x by transforming a grid of the image dimensions and interpolating x at the resulting coordinates (see Jaderberg et al. [2015] for details).

In the works presented in this thesis, we do not *optimise* transformations directly, but rather *marginalise* distributions over them and optimise those distributions' hyperparameters. Thus, we need to consider distributions

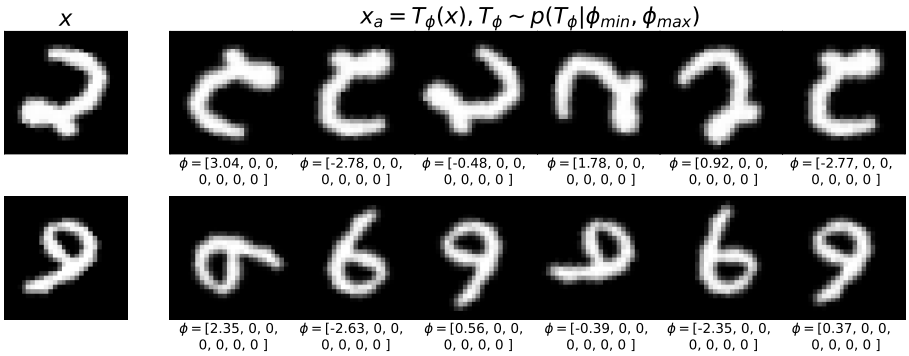


Figure 3.2: Example augmentation distribution. The learned distribution in this example is $p(T_\phi|x, \phi_{\min}, \phi_{\max}) = \mathcal{U}(\phi_{\min}, \phi_{\max})$ with $\phi_{\min/\max} \approx (\pm\pi, 0, 0, 0, 0, 0, 0)$, i.e. we learn to perform rotations, but no scale, shearing or translation. While we have chosen this augmentation distribution manually for illustrative purposes here, we will see in Sec. 3.3.3 that this is indeed the augmentation distribution we learn for rotated MNIST.

$p(T_\phi|x)$ that are themselves differentiable. Uniform augmentation distributions as used in Sec. 3.2.4 and Sec. 3.3 are parameterised as

$$p(T_\phi|x) = p(T_\phi|x, \phi_{\min}, \phi_{\max}) = \mathcal{U}(\phi_{\min}, \phi_{\max}). \tag{3.9}$$

To obtain differentiability with respect to $\phi_{\min/\max} \in \mathbb{R}^7$ we sample via the reparametrisation trick [Kingma and Welling, 2014]

$$\phi = \phi_{\min} + (\phi_{\max} - \phi_{\min})\varepsilon,\ \varepsilon \sim \mathcal{U}(0, 1). \tag{3.10}$$

This means that for different datasets, different augmentation distributions are learned by estimating $\phi_{\min}, \phi_{\max}$ per dataset. For example, on rotated MNIST (see Sec. 3.3.3) we might learn $\phi_{\min/\max} \approx (\pm\pi, 0, 0, 0, 0, 0, 0)$ corresponding to full rotational invariance (sampling any angle between $-\pi$ and π) but no scaling, shearing or translations. Samples $x_a = T_\phi(x)$ arising from such a transformation distribution are visualised in Fig. 3.2. Note that in this setting, $p(T_\phi|x)$ is relatively simple and fully described by the 14 parameters in $\phi_{\min}$ and $\phi_{\max}$, i.e. $w_a = \{\phi_{\min}, \phi_{\max}\}$.

3.2.4 Invariant GPs

Recall that, following van der Wilk et al. [2018], we want to use GPs and their tractable marginal likelihood approximations in order to learn data augmentation/invariances. To construct an invariant $f \sim \mathcal{GP}(0, k_f(\cdot,\cdot))$ we consider functions f of a specific form, marginalising transformations as introduced in Eq. 2.6:

$$f(x) = \int g(x_a) p(x_a|x, w_a) dx_a, \tag{3.11}$$

where g is a non-invariant function.

Note the small difference in notation between Eq. 3.11 and Eq. 2.6: The augmentation distribution $p(x_a|x, w_a)$ now plays the role of the distribution over transformations $p_{w_a}(T|x)$ before. An augmented x_a can be constructed from x by applying a transformation $T(x) = x_a$, and since T is a deterministic transformation the two distributions $p(T|x)$ and $p(x_a|x, w_a)$ are equivalent. f is invariant to the augmentation distribution $p(x_a|\cdot, w_a)$ and, since Gaussians are closed under addition (also in the infinite limit), it is also a GP. Its kernel is

$$k_f(x, x') = \iint k_g(x_a, x'_a) p(x_a|x, w_a) p(x'_a|x', w_a) \mathrm{d}x_a \mathrm{d}x'_a. \tag{3.12}$$

This double integral adds additional tractability issues to the ones discussed in Sec. 3.2.2, as it is not available in closed form for any non-trivial augmentation distribution. Van der Wilk et al. [2018] develop a sample-based estimator in the case of Gaussian likelihoods,

$$\mathcal{L} = \sum_{i=1}^{N_\mathcal{D}} \mathbb{E}_{q(f(x_i))}[\log \mathcal{N}(y_i|f(x_i), \sigma^2)] - \mathrm{KL}[q(U)||p(U)] \tag{3.13}$$

$$= \sum_{i=1}^{N_\mathcal{D}} \left[-\log 2\pi\sigma^2 - \frac{1}{2}\left((y_i - \mu(x_i))^2 + \nu(x_i, x_i) \right) \right] - \mathrm{KL}[q(U)||p(U)]. \tag{3.14}$$

In the Gaussian case, only $\mu(x_i)$ and $\nu(x_i, x_i)$, the variational posterior mean and variance from Eq. 3.7, depend on the intractable kernel evaluations in Eq. 3.12. Van der Wilk et al. [2018] develop unbiased and relatively efficient estimators using samples from $p(x_a|x, w_a)$. Using those estimators they perform inference in their model. In particular, by inferring the hyperparameters w_a, they learn invariances (data augmentation) on MNIST and rotated MNIST. They do this successfully in the sense of recovering ground truth transformations as well as improving predictive performance over a non-invariant model. Thus, invariance learning via Bayesian model selection is possible in GP models.

3.3 Scaling Bayesian Model Selection to Neural Networks

This section summarises the contributions from Paper 1.

We start with a small recap: We have seen in Chapter 2 that data augmentation (or, equivalently, invariance learning) is a useful, but usually ad-hoc and hand-tuned engineering trick. When trying to *learn* a useful data augmentation scheme, we have seen that using standard losses (MSE or negative log-likelihood) is not a valid strategy (Sec. 2.2.2.1). As argued in Sec. 3.1 a promising alternative is to use the marginal likelihood. Indeed, van der Wilk et al. [2018] have shown that the marginal likelihood can be employed for invariance learning. However, as we have seen in Sec. 3.1 marginal likelihood computations are intractable for neural network models, so instead van der Wilk et al. [2018] rely on GPs. For GP models the marginal likelihood can be approximated well but they can lack predictive performance compared to modern, high capacity neural networks.

Thus, Paper 1 asks the following question: Given that marginal likelihood computations for neural networks are an active but unsolved research question (see Sec. 3.2.1 on Bayesian deep learning), can we scale the marginal likelihood approach to neural networks by using a simple approximation? Might a Bayesian *last layer* suffice for invariance learning?

3.3.1 Invariant Deep Kernel GPs

Deep kernel learning (DKL; Wilson et al. [2016a,b]) models are neural network – GP hybrid models constructed with the aim to combine the nice analytical properties of GPs (importantly, the tractable marginal likelihood computations) with the expressivity of neural networks. They are constructed by placing a neural network feature extractor $h_{w_p} : \mathcal{X} \to \mathbb{R}^d$ inside a GP covariance function. Covariance functions are closed with respect to transformations of their input, i.e. if $k_g(\cdot, \cdot)$ is a covariance function on $\mathbb{R}^d \times \mathbb{R}^d$, then $k_g(h_{w_p}(\cdot), h_{w_p}(\cdot))$ is a covariance function on $\mathcal{X} \times \mathcal{X}$. Learning the neural network weights w_p inside the kernel is done by viewing those as kernel hyperparmeters and optimising them with respect to the marginal likelihood. By using such a *deep kernel* we *marginalise* the function f but *optimise* neural network weights w_p. In other words, our model is Bayesian in the last layer only.

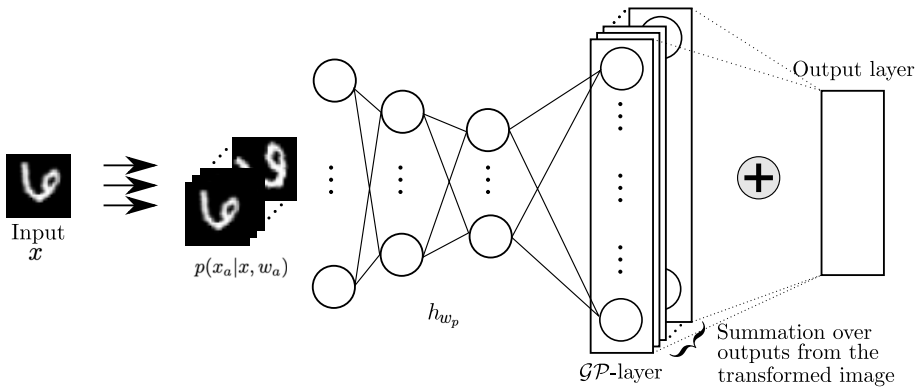


Figure 3.3: The InvDKGP architecture. For any input x, we sample from the augmentation distribution $p(x_a|x, w_a)$; each of the sample gets passed through a neural network parametrised by w_p. The last layer is a GP, which sums across sample outputs to create an invariant function. Figure from Paper 1.

Using a deep kernel in Eq. 3.11, we obtain an invariant GP with a deep kernel (InvDKGP), i.e.

$$f(x) = \int g(h_{w_p}(x_a))p(x_a|x, w_a)\mathrm{d}x_a. \tag{3.15}$$

with kernel similar to the shallow invariant kernel (3.12),

$$k_f(x, x') = \int k_g\big(h_{w_p}(x_a), h_{w_p}(x'_a)\big)p(x_a|x, w_a)p(x'_a|x', w_a)\mathrm{d}x_a\mathrm{d}x'_a. \tag{3.16}$$

The resulting model architecture is visualised in Fig. 3.3.

3.3.2 Inference in Invariant Deep Kernel GPs

Performing inference in our model corresponds to estimating the GP hyperparameters: likelihood variance σ^2, kernel variance s^2 and lengthscale ℓ (see Eq. 3.3) as well as the variational parameters Z, m, S (see Eq. 3.7) and, importantly, the invariance parameters w_a. In the deep kernel setting, we additionally learn the neural network parameters w_p. Standard deep kernel learning estimates hyper- and variational parameters jointly via marginal likelihood maximisation. In the InvDKGP model we would naively do this by

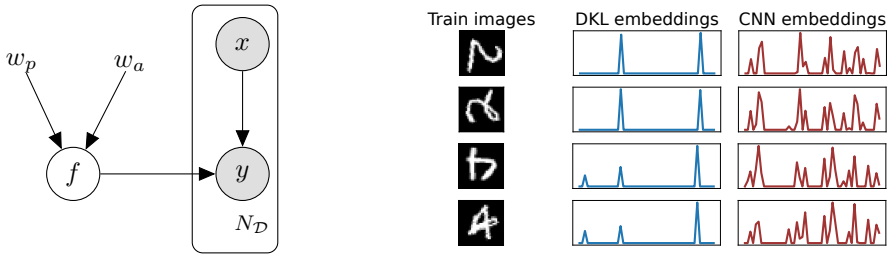


Figure 3.4: *Left:* Graphical model for the InvDKGP architecture. Both w_p and w_a are treated as model hyperparameters. For a full Bayesian treatment, we would need to marginalise the w_p along with f. *Right:* Features extracted by a DKL model (blue) and a standard neural net (red). Embeddings produced via DKL (i.e. joint training) are similar within classes: little improvement can be gained by being rotationally invariant. NN embeddings differ depending on input orientation, producing signal to learn w_a from. Figure from Paper 1.

maximising the ELBO in Eq. 3.14, replacing f with the invariant f from Eq. 3.11. There is two challenges we face when following the naive approach. In the remainder of this section, we develop solutions to those.

3.3.2.1 Overfitting and Coordinate-Ascent Training

We have argued in Sec. 2.2.2.1 that one cannot learn DA via max-likelihood and instead needs to perform type II max-likelihood estimation. I.e., one needs a hierarchical model structure such as in Fig. 3.1 and compute the marginal likelihood with respect to f_{w_p}. While InvDKGPs indeed marginalise the function f in the 'last layer', the neural network parameters in the deep kernel w_p act as kernel hyperparameters (see Sec. 3.3.1) and are not marginalised. Fig. 3.4, left, shows the graphical model of a InvDGKP. Hence, the neural network weights w_p are not protected from overfitting (in the sense of Sec. 3.1, see Ober et al. [2021] for a detailed discussion of this problem in DKL). As a consequence, joint training of w_p and w_a in the InvDKGP indeed results in overfitting. Fig. 3.4, right, shows features extracted by a deep kernel (blue) and a regular neural network (red). DKL produces overfit features in the sense that inputs from the same class are mapped to very similar activation functions, leaving us with little signal to learn invariances from.[1] The NN features are more diverse,

[1] One might argue that models which do not explicitly represent invariance but 'absorb' orientation into the features (DKL in Fig. 3.4, right) are not, a priori, less desirable. However,

providing signal for invariance learning. This insight leads us to a modified training scheme: Instead of training the GP and NN parameters jointly, we iterate between updating w_p and GP parameters in a coordinate ascent style fashion.

3.3.2.2 Correcting Model Misspecification

The choice of likelihood exacerbates the overfitting problem. As we have seen in Sec. 3.2.4, van der Wilk et al. [2018] use the Gaussian likelihood and estimate unbiased estimators for the data-dependent terms in the closed from variational expectation (3.14). However, using a Gaussian likelihood is a model misspecification for the classification problems that are considered both by van der Wilk et al. [2018] and our work (MNIST variations, PCAM, CIFAR). In the InvDKGP case, this results in collapsing augmentation distributions $w_a \to 0$ along with collapsing likelihood variance σ^2. A simple remedy is to fix σ^2 to a suitable, manually determined value. Proceeding like this and applying coordinate ascent-style training indeed enables us to successfully learn invariances in the InvDKGP (see Schwöbel et al. [2022], Sec. 5), but is an unprincipled approach that requires hand-tuning. Thus, we instead fix the model misspecification by allowing for more general likelihoods.

To do so, we develop a new lower bound to the ELBO (see Schwöbel et al. [2022], Sec. 6) that uses samples directly in a Monte Carlo estimate of Eq. 3.13 rather than relying on the closed form integral. To construct this estimator, recall that f is defined by marginalising the transformation distribution (3.11). We will omit h_{w_p} from the notation for brevity in the following). The integral is intractable but can be approximated via Monte Carlo sampling:

$$f(x) \approx \hat{f}(x) := \frac{1}{S_o} \sum_{i=1}^{S_o} g(x_a^i), \text{with } x_a^i \sim p(x_a|x, w_a). \tag{3.17}$$

We could draw multiple *sets* of orbit points — each containing S_o augmented versions of x, $\{x_a^{ji}\}_{i=1}^{S_o}$ for $j = 1, \ldots, S_{\mathcal{A}}$ — to obtain different estimators $\hat{f}$. Doing this infinitely many times, we would recover the true f (in other words, $\hat{f}$ is an unbiased estimator of f):

$$f(x) = \mathbb{E}_{\prod_{i=1}^{S_o} \cdot p(x_a^i|x)} \left[\hat{f}(x) \right] =: \tilde{\mathbb{E}} \left[\hat{f}(x) \right]. \tag{3.18}$$

Here, $\prod_{i=1}^{S_o} p(x_a^i|x)$ is the product density over S_o orbit densities.

as we will see in Table 3.1, models that explicitly incorporate invariance perform a lot better in our experiments (e.g., comparing M7 with M9). Additionally, modelling invariance explicitly has qualitative advantages such as better interpretability and disentangled representations.

Note that f is stochastic in x but deterministic in g, which is a GP. Thus, we can rewrite the expectation over f as an expectation over g,

$$\mathbb{E}_{q(f(x))}[\log p(y|f(x))] = \mathbb{E}_{q(g)}[\log p(y|f(x))] \tag{3.19}$$

$$= \mathbb{E}_{q(g)}\left[\log p\left(y\middle|\tilde{\mathbb{E}}[\hat{f}(x)]\right)\right] \tag{3.20}$$

$$\geq \mathbb{E}_{q(g)}\left[\tilde{\mathbb{E}}\left[\log p\left(y\middle|\hat{f}(x)\right)\right]\right]. \tag{3.21}$$

Here, we have obtained the last line (3.21) using Jensen's inequality, which we can do whenever the likelihood is *log-concave* in f. This holds for many common likelihoods, e.g. Gaussian and Softmax.

The bound (3.21) is tight when $\mathrm{Var}(\hat{f}(x)) = 0$, it becomes tighter as S_o increases (similar to Burda et al. [2016]). Thus, aggressive sampling recovers accurate variational inference. We can estimate the right-hand side of Eq. 3.21, *without bias*, as

$$\frac{1}{S_g}\sum_{k=1}^{S_g}\frac{1}{S_{\mathcal{A}}}\sum_{j=1}^{S_{\mathcal{A}}}\log p\left(y\middle|\frac{1}{S_o}\sum_{i=1}^{S_o}g_k(x_a^{ji})\right). \tag{3.22}$$

As before, S_o is the number of orbit samples. $S_{\mathcal{A}}$ is the number of *sets* of samples (or $\hat{f}$) that we draw, and S_g is the number of times we draw from the GP in order to compute the variational expectation as per usual. Recall that we need to sample extensively in order to keep the bound above tight, so it is important to do so efficiently. This can be done by sampling the approximate posteriors $q(g)$ using Matheron's rule [Wilson et al., 2020]. By doing so, sampling S_g GPs is cheap compared to sampling from the orbit.In practice, we choose large S_o to obtain $\mathrm{Var}(\hat{f}(x)) \approx 0$, and so it is sufficient to choose $S_{\mathcal{A}} = 0$.

Replacing the Gaussian variational expectation in Eq. 3.13 , we obtain the stochastic ELBO:

$$\frac{1}{S_g}\sum_{k=1}^{S_g}\frac{1}{S_{\mathcal{A}}}\sum_{j=1}^{S_{\mathcal{A}}}[\log p(y|\frac{1}{S_o}\sum_{i=1}^{S_o}g_k(x_a^{ji}))] - \mathrm{KL}[q(U)||p(U)]. \tag{3.23}$$

The new bound allows us to use arbitrary log-concave likelihoods, in particular the softmax likelihood appropriate for classification. The benefits of the sample based bound are three-fold: We broaden model specification, avoid using the hand-tuned Gaussian likelihood variance, and double training speed (see Paper 1, Sec. 6.1).

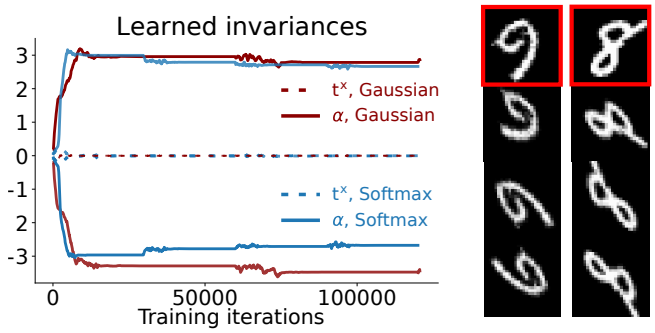


Figure 3.5: *Left*: Learned invariance parameters (rotation α in radians and x-translation t_x) for rotMNIST. The Gaussian and Softmax model both learn to be almost fully rotationally invariant (i.e. $\alpha_{\text{min/max}} \approx \pm\pi$), but the Softmax model learns faster (it takes fewer iterations and is faster per iteration, see Paper 1, Sec. 6.1). The models learn *not* to be invariant w.r.t. translation (i.e. $t^x_{\text{min/max}} \approx 0$). *Right*: Two training images (red frames) and samples from the learned $p(x_a|x, w_a)$.

3.3.3 Experimental Results

Using these two modifications to the naive, joint-training approach (modified training scheme and the new bound allowing for softmax likelihood) we successfully learn invariances using the InvDKGP. We show this on variations of the MNIST datasets as well as a medical example (PCAM, Veeling et al. [2018]). Fig. 3.5 shows the invariances we learn on rotated MNIST. The dataset is generated by randomly rotating MNIST images by an arbitrary angle, so the ground truth rotational invariance corresponds to $\pm\pi$ radians. The InvDKGP recovers this rotational invariance as is shown in Fig. 3.5, left. Fig. 3.5 also contains samples from the augmentation distribution $p(x_a|x, w_a)$. As expected the InvDKGP outperforms both non-invariant models as well as the invariant GP with a shallow kernel (see Table 5.1).

In a last experiment, we systematically explore the limitations of our method. In the previous experiments we have considered h_{w_p} to be relatively simple neural networks. For the reasons discussed in Sec. 3.3.2.1, we have trained NN and GP parameters iteratively. Thus, we have essentially pre-trained feature extractors h_{w_p} on the unaugmented dataset, learned invariances, fine-tuned the feature extractors according to the learned invariances, and so on. We now try to apply this training scheme on CIFAR-10, with a ResNet-18-based [He et al., 2016] h_{w_p}, and find that we cannot learn invariances in this setting. We investigate why this is the case by experimenting with different levels of

	Model	Likelihd.	Test acc.
M1	NN	Softmax	0.9433
M2	Non-inv. Shallow GP	Gaussian	0.8357
M3	Non-inv. Shallow. GP	Softmax	0.7918
M4	Inv. Shallow GP	Gaussian	0.9516
M5	Inv. Shallow. GP	Softmax	0.9316
M6	Non-inv. Deep Kernel GP	Gaussian	0.9387
M7	Non-inv. Deep Kernel GP	Softmax	0.9351
M8	InvDKGP	Gaussian	**0.9896**
M9	InvDKGP	Softmax	0.9867

Table 3.1: Rotated MNIST, test accuracies. Invariant models outperform their non-invariant counterparts, deep kernels outperform shallow ones. InvDKGP perform best, outperforming state-of-the-art of 0.989 for learned invariance on this dataset [Benton et al., 2020]. Table from Paper 1.

augmentations during pre-training of h_{w_p} as well as in the GP layer (see Paper 1, Sec. 7 for details). We find that complex feature extractors cannot be pre-trained sufficiently well without data augmentation (i.e. invariances), and adding invariances later does not improve performance in this case. Models where h_{w_p} is pre-trained *already with the right invariances* do significantly better. Thus, the iterative training procedure that we have developed in Sec. 3.3.2.1 in order to overcome the overfitting problem of our "partly Bayesian" model does not work to our benefit in this case. A full Bayesian treatment, i.e. marginalising *all* weights might be unavoidable in certain cases in order to learn invariances.

3.4 Summary

We have seen in Chapter 2 that data augmentation (or, equivalently, invariances) cannot be learned using standard losses such as MSE or negative log-likelihood. As a consequence, we have considered van der Wilk et al. [2018]'s approach of phrasing invariance learning as a Bayesian model selection problem in Sec.3.2.4. They do so for GP models, where the marginal likelihood — the quantity needed to perform Bayesian model selection — is available in closed form. In Sec. 3.3.1 (Paper 1) we have investigated a way to extend this approach from GPs to neural networks: using DKL-based invariant models which are Bayesian in the last layer only. To train our models we needed to overcome overfitting problems (via iterative training) and correct model misspecification (via our new bound which allows for the use of arbitrary log-concave likelihoods). In combination, this enabled us to successfully learn invariances on MNIST variations and a

medical imaging dataset. On CIFAR-10, however, due to the need for a more complex model architecture, being partly Bayesian does not suffice for invariance learning. We conclude that phrasing invariance learning as a model selection problem is a promising approach also in neural networks, but to develop it to its full potential we depend on better marginal likelihood approximations.

Chapter 4

Invariance Learning via Pose-Normalisation

In Chapters 2 and 3 we have constructed invariant models by marginalising over augmentation (or transformation) distributions,

$$p(y|x) = \int p(y|T(x))p(T|x)\ \mathrm{d}x. \tag{4.1}$$

Investigating this equation we might consider another approach to arriving at invariance, *optimising* instead of *marginalising* the transformations T. We could model

$$p(y|x) = \max_T p(y|T(x)). \tag{4.2}$$

In practice, this would correspond to transforming x such that it can optimally be classified — essentially pose-normalising the input. This approach to achieving invariance is implemented in Spatial Transformer Networks (STNs, Jaderberg et al. [2015]).

4.1 Spatial Transformer Networks

Spatial transformer networks consist of two parts, a localisation and a classification network. The localiser computes the optimal pose for the

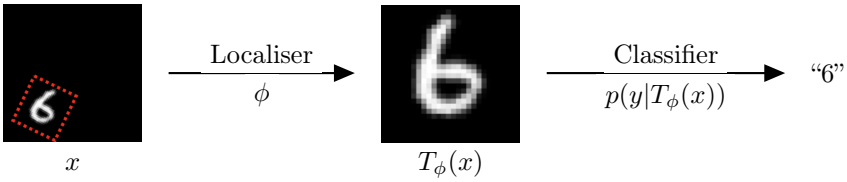


Figure 4.1: A spatial transformer network consists of two parts. The localiser network estimates and applies the optimal transformation ϕ ($\phi = [0.2\pi, 1.4, 1.4, 0, 0, 0, 0]$ in this example) and the classifier network performs the downstream task on the transformed image.

downstream task at hand. It does so by estimating transformation parameters ϕ and applying the corresponding transformation T_ϕ (see Fig. 4.1). The classification [1] network solves the downstream task, computing $p(y|T_\phi(x))$. Classifier and localiser are parametrised by neural networks, and are trained jointly via gradient updates. To do so, the transformations have to be parametrised in a differentiable manner as described before (Sec. 3.2.3). Jaderberg et al. [2015] extensively use affine transformations similar to the ones discussed in Sec. 3.2.3 and other relatively simple classes of transformations such as thin-plate splines. Detlefsen et al. [2018] show that the approach also works for more expressive diffeomorphic transformations.

Spatial transformer networks can produce (approximately) invariant functions if all input images are mapped to the same canonical orientation before passed on to the downstream model. This allows STN models to improve predictive performance [Jaderberg et al., 2015, Detlefsen et al., 2018]. They can also be useful in an interpretability context: Jaderberg et al. [2015] train an STN with multiple localisers to classify different bird species on the CUB-200-2011 dataset. They find that one of the localisers focuses on identifying the head, one on the body, and so on. We were, however, unable to reproduce this experiment and find that, in general, STNs can be hard to optimise. The next section is concerned with improving this.

[1] In principle, this could be any downstream task, so 'predictor network' might be a better term. Jaderberg et al. [2015] as well as our extension which will be discussed in Sec. 4.2 consider downstream classification tasks so we will refer to this part of the model as the 'classifier'.

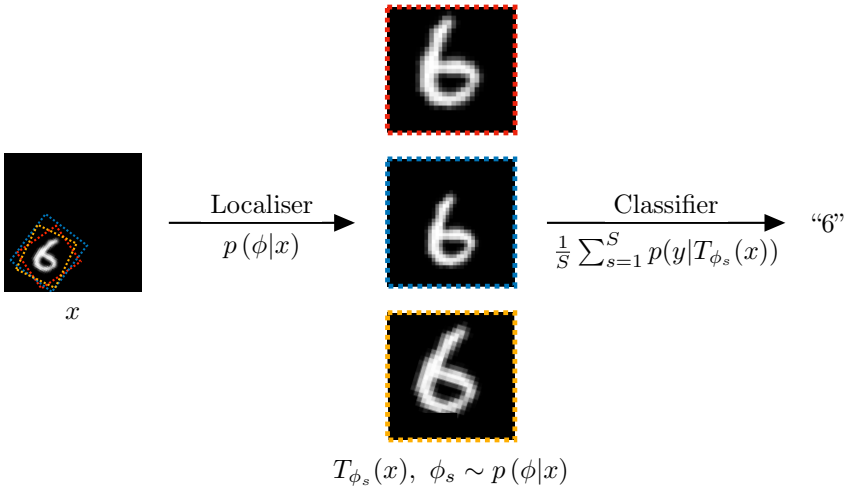


Figure 4.2: Probabilistic spatial transformer network. Similar to a regular spatial transformer network, its probabilistic extension consists of two parts. The localiser network of a P-STN, however, estimates a *distribution* over transformation parameters $p(\phi|x)$. S different transformations are sampled and applied. The resulting transformed images are fed through the classifier network whose predictions are averaged in order to approximate the variational expectation (4.14). Note how applying the different samples from the transformation distribution results in augmenting the data around the mean transformation.

4.2 A Probabilistic Extension to Spatial Transformer Networks

This section summarises the contributions from Paper 2.

In practice, STNs can be brittle and difficult to train. If the localiser predicts a wrong transformation T_ϕ (for example, by zooming in on a corner such that only black background is visible in the image in Fig. 4.1) we might lose any signal for downstream task and backpropagation. This problem is exacerbated for non-invertible T_ϕ [Detlefsen et al., 2018], but is also present in other cases. Secondly, the STN modelling assumption itself can be challenged: Is there really one 'true' underlying pose that images should be normalised to, especially since we know

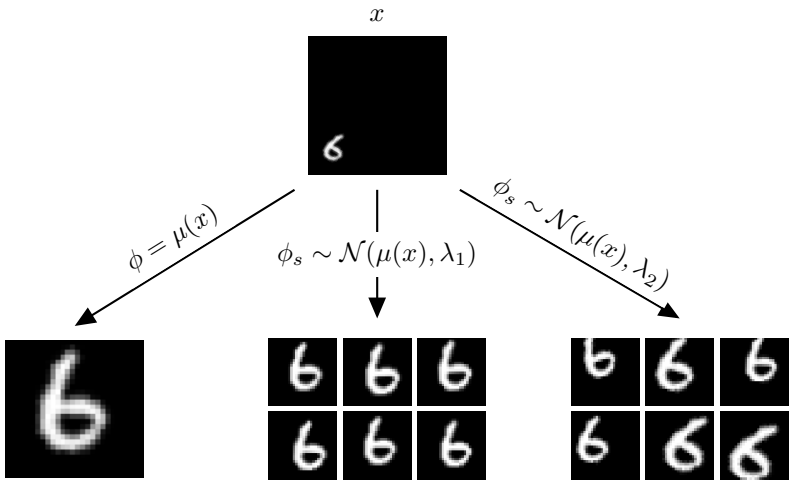


Figure 4.3: An input image x and its example transformations $T_\phi(x)$ (bottom row). The leftmost transformation is the mean transformation. The middle transformations are drawn from the normal with large precision λ_1 corresponding to augmenting little around the mean transformation $\mu(x)$. The transformations on the right are drawn from a distribution with smaller precision λ_2, corresponding to augmenting around the mean more aggressively.

that the small variations introduced by standard data augmentation (Ch. 2) are immensely helpful for training? Paper 2 is a reply to these two concerns. Our probabilistic extension to the STN (P-STN) estimates an optimal mean transformation μ while taking its uncertainty into account, i.e. also modelling a precision λ. This transformation uncertainty we marginalise over as before (4.1), effectively 'getting to try out' different transformations. We will see that this is equivalent to applying a type of localised data augmentation.

4.2.1 Model

The difference to previously discussed models is as follows: In the InvDKGP in Sec. 3.3.1 we consider *uniform, global* augmentation distributions $p(T|x) = p(T) = \mathcal{U}(\phi_{\min}, \phi_{\max})$. The P-STN employs *normal, per-image* augmentation distributions $p(T|x) = \mathcal{N}(T|\mu(x), \lambda)$. We will infer the augmentation distribution via amortised variational inference, i.e. the parameters of the augmentation distribution are modelled via neural networks

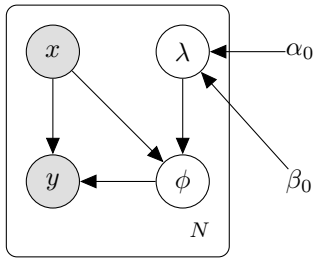


Figure 4.4: Graphical model, visualisation adapted from Paper 2. Grey nodes are observable and white are latent.

like in the deterministic STN.

Here, λ denotes the precision of the transformation distribution. When marginalising over the transformation distribution, a small precision λ corresponds to trying many different transformations, or augmenting the data aggressively (Fig. 4.3, right). A large precision λ implies little uncertainty around the mean transformation, or not augmenting the (pose-normalised) image much (Fig. 4.3, middle). We know from Sec. 2.2.2.1 that naively trying to infer λ will result in a collapsing augmentation distribution. We thus introduce a prior on λ, resulting in a model structure as visualised in Fig. 4.4. The model thus factorises as

$$p(y, x, \phi, \lambda) = p(y|x, \phi)p(\phi|\lambda, x)p(\lambda)p(x). \tag{4.3}$$

The input data density $p(x)$ does not depend on model parameters λ and θ, thus it can be specified without affecting the model. The prior over λ is a Gamma distribution, with the λ_i per observation assumed to be independent, $p(\lambda) = \prod_{i=1}^{N_\mathcal{D}} p(\lambda_i)$ with density

$$p(\lambda_i) = \Gamma(\alpha_0, \beta_0). \tag{4.4}$$

We choose the Gamma prior due to its conjugacy with the normal likelihood $p(\phi|\lambda, x) = \mathcal{N}(\phi|\mu(x), \lambda)$, this is generally considered a robust setup for estimating variances [Stirn and Knowles, 2020, Takahashi et al., 2018, Detlefsen et al., 2019]. We wish to perform inference over the latent variables by maximising

$$\log p(x, y) = \log \iint p(x, y, \phi, \lambda)\ \mathrm{d}\phi\ \mathrm{d}\lambda. \tag{4.5}$$

4.2.2 Inference in Probabilistic Spatial Transformer Networks

The marginal likelihood (4.5) is intractable, and so is the posterior $p(\phi, \lambda|x, y)$. Thus, we choose a variational approximation

$$q(\phi, \lambda) := p(\phi|\lambda, x)q(\lambda). \tag{4.6}$$

Where $p(\phi|\lambda, x) = \mathcal{N}(\phi|\mu(x), \lambda)$ as before and $q(\lambda) := \prod_{i=1}^{N} \Gamma(\alpha_i, \beta(x_i))$. Here β is a neural network, i.e. we use amortised inference in a similar way to the VAE model [Kingma and Welling, 2014]. We choose constant $\alpha_i = 1$.

Using Jensen's inequality, we obtain the following evidence lower bound

$$\begin{aligned}
\log p(y, x) &= \log \iint p(y, x, \phi, \lambda)\ \mathrm{d}\phi\ \mathrm{d}\lambda && (4.7)\\
&\geq \iint \log\left(\frac{p(y, x, \phi, \lambda)}{q(\phi, \lambda)}\right) q(\phi, \lambda)\ \mathrm{d}\phi\ \mathrm{d}\lambda && (4.8)\\
&= \iint \log\left(\frac{p(y|x, \phi)p(\lambda)p(x)}{q(\lambda)}\right) p(\phi|\lambda, x)q(\lambda)\ \mathrm{d}\phi\ \mathrm{d}\lambda \\
&= \underbrace{\mathbb{E}_{q(\phi,\lambda)} \log p(y|x, \phi)}_{\text{classification loss}} + \log p(x) - \mathrm{KL}(q(\lambda)\|p(\lambda))\,. && (4.9)
\end{aligned}$$

Under our modelling assumptions, we can compute the classification loss (i.e. the variational expectation) like so

$$\begin{aligned}
\mathbb{E}_{q(\phi,\lambda)} \log p(y|x, \phi) &= \iint \log p(y|x, \phi)q(\phi, \lambda)\ \mathrm{d}\phi\ \mathrm{d}\lambda && (4.10)\\
&= \iint \log p(y|x, \phi)p(\phi|\lambda, x)q(\lambda)\ \mathrm{d}\phi\ \mathrm{d}\lambda && (4.11)\\
&= \int \log p(y|x, \phi) \int \mathcal{N}(\phi|\mu(x), \lambda)\Gamma(\lambda|\alpha, \beta(x))\ \mathrm{d}\lambda\ \mathrm{d}\phi && (4.12)\\
&= \int \log p(y|x, \phi) t_{2\alpha}(\phi|\mu(x)), \tfrac{\beta(x)}{\alpha})\ \mathrm{d}\phi. && (4.13)
\end{aligned}$$

Here t denotes a scaled and location-shifted Student's t-distribution with mean $\mu(x)$, scaling β, and 2α degrees of freedom. It arose from marginalising over λ in $q(\phi, \lambda)$, and is the distribution that we draw samples from in our

implementation. $p(y|x,\phi) = p(y|T_\phi(x))$ is what the classifier will compute. In practice, we approximate Eq. 4.13 using an unbiased estimate

$$\begin{aligned} &\mathbb{E}_{q(\phi,\lambda)} \log p(y_i|x_i,\phi_i) \approx \frac{1}{S}\sum_{s=1}^{S} \log p(y_i|x_i,\phi_{i,s}), \\ &\text{with } \phi_{i,s} \sim t_{2\alpha_i}(\cdot|\mu(x_i),\alpha_i,\beta(x_i)) \end{aligned} \tag{4.14}$$

and backpropagate through neural networks $\mu(x)$ and $\beta(x)$ with the reparametrisation trick.

Plugging Eq. 4.14 into Eq. 4.13 we obtain the final ELBO

$$\mathcal{L}_{p,q}(x,y) \approx \sum_{i=1}^{N} \frac{1}{S}\sum_{s=1}^{S} \log p(y_i|x_i,\phi_{i,s}) - \mathrm{KL}\left(q(\lambda)||p(\lambda)\right) + \text{const}, \tag{4.15}$$

where the constant term corresponds to the log-prior over inputs $\log p(x)$. Since this term does not depend on any of the model parameters, we can disregard it during inference, where we maximise ELBO (4.15) using any gradient-based method. The KL-divergence between two Gamma distributions is analytically tractable and differentiable.

In our experiments we follow Higgins et al. [2016] and introduce a weight parameter w to the KL-term. This way, we need to tune w but in turn our model becomes robust to the choice of prior — otherwise we would need to tune the prior directly via β_0. We perform a grid-search on a validation set to find the optimal w and choose $\alpha_0 = \beta_0 = 1$ for all experiments. Our model is similar to the model in Sec. 3.3.1 in that it implies marginalisation, and thus data augmentation, at *test-time* as well as the usual training time. At test time, we draw $S = 10$ transformation samples. At training time, we find that sometimes $S = 1$ sample suffices as suggested by Kingma and Welling [2014], other times we obtain better results with $S = 10$ also at training time. We will state the number of samples drawn in the individual experiments.

4.2.3 Experimental Results

We investigate the P-STN along three dimensions: the localisation task, the classification task and model calibration. The following section contains a high-level summary of the experiments, see Paper 2 for experimental detail.

4.2.3.1 Marginalising transformations improves localisation accuracy

STN models are trained end-to-end, i.e. the transformation is learned based on the labels for the downstream task alone. We do not have label information available for the transformations. This property is useful since we can do with sparsely labelled data, but it makes STNs hard to fit. The experiments in this section investigate whether the probabilistic formulation improves localisation. We study whether estimating a posterior over transformations and marginalising, i.e. 'getting to try multiple transformations', makes the task easier.

Rotated MNIST is revisited in the first experiment of this section. We generate the dataset by randomly rotating the original MNIST digits by r radians. We store the applied rotations in order to have ground truth transformations available, $\phi_{\text{true}} = [r_{\text{true}}, 0, 0, ..]$. We set up our experiment as follows: Firstly, we train a CNN on regular MNIST, i.e. an already pose-normalise dataset. We copy its weights into the classifier model of a STN and P-STN model. With those classifier weights frozen, we then train STN and P-STN localisers, effectively learning to recover and 'undo' the applied transformations ϕ_{true}. Training localiser and classifier separately in this manner has two advantages. Firstly, it allows us to evaluate localiser performance disentangled from the downstream task (see Fig. 4.5). Secondly, the training procedure avoids a typical failure mode of STNs that we will investigate later (Sec. 4.2.3.4).

The pretrained CNN obtains 99.4% test accuracy on MNIST. On rotated MNIST accuracy drops to 41.2% with frozen weights (i.e. no retraining). Both the STN and P-STN ($S = 10$ training samples, $w = 3 \cdot 10^{-5}$) both pose-normalise successfully — they recover the ground truth transformations to a satisfactory degree. When training the localisers with classifier weights frozen as described above, the STN test acc. is 76.13% and the P-STN test acc. 82.98%. We compute the expected average transformation error on the $N = 10k$ rotated MNIST test images as

$$\varepsilon = \frac{1}{N} \sum_{i=1}^{N} \|\phi_{\text{true}}(x_i) - \mu(x_i)\| \mod \pi. \tag{4.16}$$

The resulting values are $\varepsilon = 0.76$ for the STN and $\varepsilon = 0.59$ for the P-STN. The P-STN outperfoms the STN, i.e. marginalising transformation and accounting for transformation uncertainty helps with the localisation task.

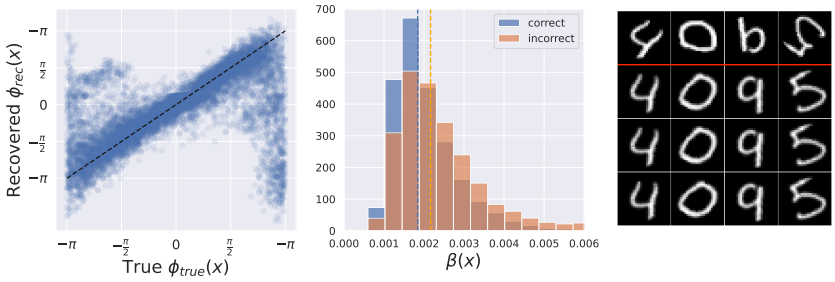


Figure 4.5: Rotated MNIST experiment, figure adapted from Paper 2. *Left:* True transformation (rotation angles in radians) against learned transformations (mean). The P-STN generally recovers the ground truth transformations well. *Middle:* Variance network outputs. When the transformation recovery is poor (the error ε is above the median, in orange) the variances are slightly higher than when the localisation works well (blue). *Right:* Images from the dataset (top row) and samples from the P-STN localiser (bottom three rows), those appear pose-normalised.

Random Placement Fashion MNIST and Mapillary Street Signs experiments yield similar results: As in the rotated MNIST example, marginalising transformations helps with the localisation task on a modified version of fashion MNIST (images are randomly placed on a black background, $\phi_{\text{true}} = [0, \cdots, 0, t^x_{\text{true}}, t^y_{\text{true}}]$) and the more challenging Mapillary Street Sign dataset. Notably, on this complex real-world dataset we demonstrate that the P-STN is compatible with more complex model architectures (ResNet18) and that it significantly improves performance: it achieves 92.2% test accuracy, compared to 76.0% for a standard ResNet18 and 90.6% for a deterministic STN. See Paper 2 for details.

4.2.3.2 Marginalising Transformations Improves Classification Accuracy

Recall that the variational expectation for the P-STN (4.13) is an integral over transformations $p(\phi|x)$. As such, it reminds us of the principled model for data augmentation as formulated in Eq. 2.6 and Eq. 4.1. Indeed, while the mean transformation $\mu(x)$ computes the pose-normalisation, the uncertainty in $p(\phi|x) = t_{2\alpha}(\phi|\mu(x)), \frac{\beta(x)}{\alpha})$ and the slightly different samples we draw when computing the MC approximation (4.14) correspond to localised, per-image data augmentation (see Fig. 4.2). We hypothesise that this data augmentation should

Figure 4.6: *Left:* Performances of P-STN, STN and CNN on MNIST subsets (mean ± one STD across five folds). *Right:* The P-STN learns to localise traffic signs in the challenging MTSD dataset. 10 sampled transformations are shown with their corresponding bounding boxes overlaid, here on test data. The learned variations improve performance on the downstream classification task. Figure from Paper 2.

help with the downstream classification task.

Small subsets of MNIST provide a useful test bed for this hypothesis for two reasons: Firstly, data augmentation is usually particularly important for small datasets ($N_{\mathcal{D}} \ll N_{w_p}$ in Eq. 2.1), so using small datasets should be an ideal setup to evaluate a data augmentation strategy. Secondly, since MNIST data is already pose-normalised the mean transformation should have little impact, and so any possible benefits should stem from the uncertainty in the transformation distribution and the resulting data augmentation we apply. Indeed, we find that the P-STN outperforms CNN and STN on small MNIST subsets by large margins (Fig. 4.6, left). On larger datasets the advantage of data augmentation diminishes as expected.

UCR Time Series are investigated in the next experiment of this section. The experiment illustrates that the class of affine transformations for T can easily be replaced with any other differentiable class of transformations. For this time series dataset we use the diffeomorphic transformations from Detlefsen [2018]. We note that for this family of transformations the parameters are much less interpretable than in the affine case, so *learning* over hand-crafting a data augmentation scheme is particularly useful. See Fig. 4.7, left for such a learned time series augmentation. As before, P-STN outperforms or is on par with the

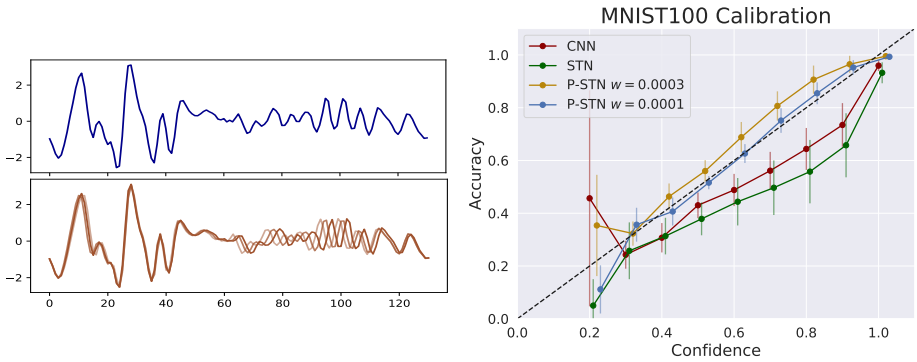


Figure 4.7: *Left:* Examples augmentations for a time-series from the FaceAll dataset. The top plot shows the original time-series and the bottom plot shows augmented samples. *Right:* The different models' calibrations on MNIST100. Figure from Paper 2.

other models on four out of five of the dataset we use. It performs less well than a standard CNN on one of the datasets (here, the vanilla CNN already achieves near-perfect test accuracy at 99.63%).

4.2.3.3 Marginalising Transformations Improves Model Calibration

We have seen that the uncertainties in the per-image transformation distributions were somewhat meaningful (Sec. 4.2.3.1) — images that are harder for our model to localise are associated with larger transformation uncertainties. We will now investigate whether those meaningful localisation uncertainties translate into meaningful uncertainties in the downstream task, i.e. we will study classifier calibration.

At test-time, we evaluate the 'actual' probabilities rather than their logarithm,

$$p(y|x) = \int p(y|x, \phi)q(\phi)\ \mathrm{d}\phi \approx \frac{1}{S}\sum_{s=1}^{S} p(y|T_{\phi_s}(x)), \tag{4.17}$$

and we investigate whether the uncertainty in $p(y|x)$ corresponds to the quality of predictions. This is visualised in the calibration plot in Fig. 4.7 (right panel) for the MNIST100 subset classification task.We plot calibration for the CNN, STN and for P-STNs with two different w-parameters; $w = 0.0003$ yields optimal performance and $w = 0.0001$ yields optimal calibration. Following Guo et al. [2017], Küppers et al. [2020] and Küppers et al. [2021] we compute the expected

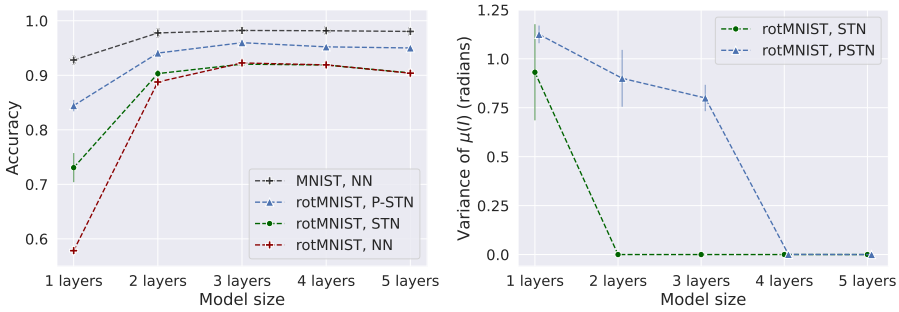


Figure 4.8: *Left:* Test accuracies for standard NN and (P-)STNs of different depths trained on rotated MNIST, as well as NN baseline on original MNIST (black). The STN (green) model does not recover the original images very well and thus behaves more like a standard NN (blue). P-STN (blue) un-transforms at least some of the rotations and is closer in accuracy to the NN on original data. *Right:* The variance of the learned transformations as a function of model depth. The STN tends to learn transformations closer to the identity (this is consistent with the test accuracies we see on the left). P-STN learns to un-transform better, at least when the classifier is simple. For bigger classifiers it falls back onto the identity transform, but performs relatively well nonetheless (see left panel). Figure from Paper 2, plotted are medians over 5 folds $\pm$ median absolute deviation.

calibration errors, those are CNN: 0.0743 ± 0.0094, STN: 0.1160 ± 0.0205, P-STN, $w = 0.0003$ (optimal performance model): 0.0567 ± 0.0065, P-STN, $w = 0.0001$ (optimal calibration model): $\mathbf{0.0271} \pm 0.0088$. The reported numbers are mean $\pm$ one STD over 5 folds. The downstream predictions are better calibrated for the P-STN than the CNN and STN models.

4.2.3.4 A Typical Failure Mode in STNs

STN are trained end-to-end, and with only label information available. Thus, one aims to learn the transformation which is optimal for solving the downstream task. Depending on the complexity of the downstream task and the classification model, it might not be necessary to transform the input at all, i.e. the downstream task might be solvable on the original input image. Indeed, this is a failure mode we observe in practice — often, the localiser simply learns the identity transform while the classifier learns to classify the

non-transformed image. As observed by other authors [Finnveden et al., 2021] using more complex classifier architectures makes the STN more prone to this failure mode. We investigate the problem in more detail in the experiment in Fig. 4.8. We start by training a neural network of different sizes on standard MNIST (black, one layer on the x-axis is `[Linear, ReLU, Dropout]`). We compare the performance of this model to (P-)STN models trained on rotated MNIST, test accuracies are plotted in the left panel of the figure. If the localisation task is performed perfectly, the (P-)STN models should be able to recover the same accuracy as the model trained on the original, non-rotated dataset. In the right panel, we plot the variance of (mean) transformations learn by the (P-)STN models. We report medians over 5 runs, with error bars corresponding to one median absolute deviation. Values close to 0 indicate that the localiser does not transform the image, i.e. it learns the identity transform. Larger values indicate that the localiser learns transformations. As hypothesised, for larger classifiers the localisers do not transform the images. Due to the increased capacity of the model, we nonetheless achieve decent classification accuracies (left panel). The P-STN learns to localise the rotated images successfully (large variance in the right panel, and high accuracy on the left), at least for smaller classifier sizes. The STN does not localise the images as well, and performs similar to a standard neural net in most runs. We conclude that thanks to it 'trying out multiple transformations', the P-STN avoids this failure mode to an extend. We also note that this property, while useful, is somewhat orthogonal to our interest in this work, and we have avoided the failure mode in the previous experiments of this section by considering models with *fixed*, pretrained classifiers.

4.3 Summary

Compared to the invariant deep kernel GP models (Sec. 3.3) spatial transformer networks (Sec. 4.1) take, in a sense, the opposite approach to invariance learning. They optimise instead of marginalising transformations, hereby pose-normalising inputs before solving downstream tasks. In this chapter, we have introduced the P-STN, a probabilistic extension to spatial transformer networks. Compared to their deterministic counterparts, the benefits are three-fold. Firstly, the P-STN improves localisation accuracy as we have demonstrated on (Fashion) MNIST variations and the Mapillary street sign dataset (Sec. 4.2.3.1). Secondly, it improves accuracy in downstream classification tasks as we have demonstrated on MNIST subsets and the UCR time series datasets (Sec. 4.2.3.2). The mechanism behind these improvements is the per-image, localised data augmentation scheme that is implied by the P-STN (see Fig. 4.2). Lastly, the P-STN achieves improved

calibration in the downstream task (Sec. 4.2.3.3) compared to its deterministic counterpart as well as a standard CNN classifier without localisation. Having discussed data augmentation and invariance learning in the previous chapters, we will now turn to the fairness question. Can the developed methods help us de-bias algorithms as hypothesised in Chapter 1?

Chapter 5

Data Augmentation for Bias-Correction

Much of the work in this chapter is inspired and informed by discussions with students in the DTU course on Deep Learning in 2019 and 2020. Big thanks in particular to Zineb Fadili, Victor Célérier, Riccardo Ricci, Paul Romieu, Ida Villumsen , Line Vognsen, Nanna Markers, Aleksander Oliver Pratt-Dam, Charlotte Friis Theisen, and Martin Johnsen. This chapter contains original work, i.e. work that is not included in Papers 1 - 3.

In Chapter 1 we have presented the idea that algorithmic bias often stems from data bias. If this is so, might data augmentation and the techniques developed in Chapter 2-4 help to reduce such biases by increasing the data set quality? This is the question we will explore in this chapter.

We will study the CelebA dataset [Liu et al., 2015] containing roughly 200k images of celebrity faces. The images are annotated with demographic and other attributes, which makes them useful for our investigation of demographic bias. In particular, we will analyse whether and how the celebrities' age is associated with whether or not they are considered attractive by human and machine annotators, i.e. whether there is an age bias present in the data and, consequently, in our modelling thereof. We will then investigate whether data augmentation might alleviate such bias.

To make it explicit: We will build a machine learning model to predict whether or not someone can be considered 'attractive'. This seems like a task both trite and eerie — like many other fairness-related examples, one might feel that it is one that should not be automated in the first place. However, it is already a reality. The popular dating app Tinder has used algorithms to score people's attractiveness [Arch, 2020]. The controversial ELO score has since been deprecated [Carman, 2019], but Tinder and other dating apps continue to rely on algorithms to suggest relevant matches to their users hereby reflecting and reproducing societal norms.

Whether your society considers you attractive has huge impacts on your live including, but not limited to, the personal domain: We tend to associate positive traits such as being happier, more successful or making better friends or partners with attractiveness [Dion et al., 1972]. In the professional domain, this stereotype seems to translate into a bias towards hiring more attractive candidates [Shahani-Denning, 2003], as well as attractive employees earning higher wages [Pfeifer, 2012].

What is beautiful is a cultural norm that is ever-changing. One important player in this process of defining beauty standards is the fashion industry, which has lately been seeing efforts towards more diversity. Campaigns feature more plus-size models, people of color, non-binary and older individuals than in the past [nyt, 2021, the, 2020].

Exposing people to such different beauty standards has the potential to create cultural change. Thus, a celebrity dataset collected in the future, might reflect a more diverse view on what is attractive than CelebA collected in 2015. Norms change, the world changes, and, as a consequence, datasets change. Thinking about such dynamics it becomes apparent that a fairness model might need to be able to map the downstream effects of fairness interventions rather than simply evaluating a single decision in isolation. We will investigate what such a long-term modelling strategy might look like in Chapter 6. For now, let us return to the standard, static setting: We have pre-collected, academic benchmark dataset, train an algorithm and then aim to investigate the algorithm's fairness. To do so, we will evaluate algorithmic bias according to the following two metrics which are influential in the fairness literature. [1]

Definition 5.1 (Fairness metrics) Let A be a protected demographic attribute, $\widehat{Y}$ some true underlying label and Y our estimation of $\widehat{Y}$ (i.e. the model prediction). Then

[1] We note that those two are by no means the only possible metrics, and point to Chapter 6 for an extensive discussion of this.

1. *Demographic Parity* is satisfied iff.

$$P(Y = 1|A = 1) = P(Y = 1|A = 0)$$

2. *Equality of Opportunity* is satisfied iff.

$$P(Y = 1|A = 1, \widehat{Y} = 1) = P(Y = 1|A = 0, \widehat{Y} = 1)$$

In the following, A corresponds to age, i.e. $A = 1$ means an individual is labeled as 'young'. $\widehat{Y}$ corresponds to the 'attractive' label. According to the data set curators, these labels were assigned by 'a professional labeling company' [Liu et al., 2015]. Y corresponds to our model's prediction of $\widehat{Y}$, i.e. we train a model to predict the attractiveness label. For brevity, we will often refer to the individuals with $Y = 0$ as the 'old' (rather than 'non-young') population. Whether there is a concrete cut-off age defined for assigning this label is not disclosed in the original paper Liu et al. [2015].

5.1 Data Bias

Before doing any predictive modelling, we can investigate whether our training data exhibits any bias. Any such bias we expect to propagate into our model. The dataset consists of a majority of people labelled as young, $P(young = 1) = 0.7736, P(young = 0) = 0.2264$. Being young is correlated with being considered attractive, the Pearson correlation coefficient between the 'Young' and 'Attractive' attribute is 0.3877. We can evaluate the demographic parity metric with respect to our underlying data (i.e. replacing model predictions Y with true labels $\widehat{Y}$). We then find that

$$P(attractive = 1|young = 1) = 0.6173$$

and

$$P(attractive = 1|young = 0) = 0.1542.$$

Similarly, these values are 0.6031 and 0.1613 on test data. Thus, we are training out model on heavily biased data. Likely, a model trained on such data will exhibit similar biases, i.e. it likely will not satisfy demographic parity either without us intervening.

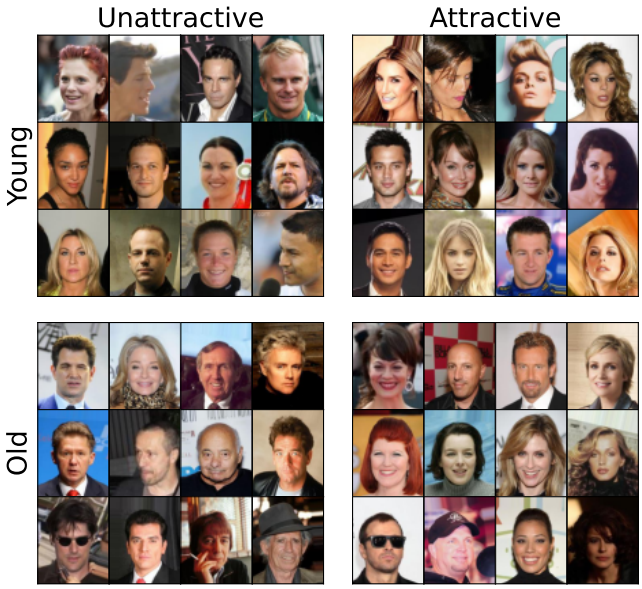


Figure 5.1: Some example images from the CelebA dataset and their labels. Columns correspond to the target label $\widehat{Y}$ and rows correspond to the protected attribute A. Investigating this random sample visually suggests that there likely are more biases in the data than the age bias we study. Not too surprisingly give cultural norms, more women than men are labelled attractive in this sample. The young woman of color in the top left ('unattractive') also seems to indicate a potential racial bias.

5.2 Model Bias

We train a model and make predictions on the test set, achieving 79.02% test accuracy. We then evaluate whether our model satisfies the fairness metrics Demographic Parity and Equalised Odds from Def. 5.1.

Demographic parity is, as expected, not satisfied. On test data, we find that

$$P(\textit{predicted attractive} = 1 | \textit{young} = 1) = 0.6687$$

and

$$P(\textit{predicted attractive} = 1 | \textit{young} = 0) = 0.2627.$$

We note that our model already makes 'friendly' predictions (somewhat over-predicting old people to be attractive), this is due to the fact that the dataset contains more young people who are more often labelled attractive. Nonetheless, demographic parity is far from satisfied.

Equality of Opportunity also is violated by our model,

$$P(predicted\ attractive = 1 | young = 1, attractive = 1) = 0.8711$$

and

$$P(predicted\ attractive = 1 | young = 0, attractive = 1) = 0.7690$$

Violating equality of opportunity means that our model amplifies the bias that is already in the data (it more often *wrongly* classifies the already less attractive part of the population as unattractive).

5.3 Debiasing the Model

We will now attempt to debias our model using our data augmentation scheme. To benchmark our approach, we will compare it to two standard techniques, namely *naive upsampling* and *adjusting thresholds* which we will detail below. These two baselines are by no means the only approaches in the vast body of fair ML and debiasing literature. Other approaches include constraint optimisation (i.e. formulating fairness requirements as part of the objective function, e.g. Dwork et al. [2012], Zafar et al. [2019]).

In order to evaluate our debiasing strategies quantitatively, we will report 'fairness ratios', i.e. the Demographic Parity ratio

$$\text{DP ratio} = \frac{P(predicted\ attractive = 1 | young = 0)}{P(predicted\ attractive = 1 | young = 1)} \tag{5.1}$$

and Equality of Opportunity Ratio

$$\text{EO ratio} = \frac{P(predicted\ attractive = 1 | young = 0, attractive = 1)}{P(predicted\ attractive = 1 | young = 1, attractive = 1)}. \tag{5.2}$$

For the original model (i.e. without debiasing), we have DPR= $\frac{0.2627}{0.6687} = 0.3929$ and EOR= $\frac{0.7690}{0.8711} = 0.8828$. For a perfectly fair model, both ratios would be equal to 1. A value larger than 1 indicates that the model is biased 'in the other direction', i.e. old individuals are classified as attractive more often.

We then choose a DP and EO rate as a minimum for our algorithm to be considered fair, say DP and EO rates $> 95\%$. For each debiasing technique, we evaluate three things: Firstly, whether this degree of fairness *can be achieved* using this method and secondly, *how much do we lose in accuracy* in order to do so. Thirdly, we will investigate *how drastic an intervention* was necessary to achieve the fairness goal under the chosen method. I.e., how much did we need to move the threshold away from the standard $t = 0.5$ (see Sec. 5.3.1) or how much did we need to upsample, respectively.

5.3.1 Adjusting Thresholds

Our chosen fairness metrics depend on the proportion of people that are (correctly) predicted attractive. A simple way to manipulate such proportions, hoping to improve their ratio, is to adjust the per-group threshold at which a person is predicted attractive [Hardt et al., 2016, Barocas et al., 2017]. In a binary classification problem, the natural hence usually unstated threshold if 0.5, i.e. we consider an individual to be predicted attractive if $P(attractive) \geq t$ with $t = 0.5$. By setting t to lower values we can make it 'easier' to be classified attractive. In particular, we can set two different thresholds $t_{A\in\{0,1\}}$ for the minority and majority groups, respectively. This way, we manipulate EO and DP ratios in a straightforward manner.

Note that this approach is a post-hoc recalibration technique, i.e. it does not require retraining the model. It does, however, require access to the protected attribute at test-time in order to determine which prediction threshold $t_{A=0}$ or $t_{A=1}$ to apply. Figure 5.2 shows the accuracy, the total proportion of people predicted attractive as well as DP and EO ratios, all as a function of different thresholds for the minority group $A = 0$. For consistency with Fig. 5.3 and 5.4 we plot $1 - t_{A=0}$ on the $x-$axis, such that larger values indicate more extreme fairness interventions. As expected, decreasing the threshold $t_{A=0}$ (equivalently, increasing $1 - t_{A=0}$) increases the proportion of people classified as attractive (grey curve), resulting in arbitrarily good EO (green) and EO (orange) ratios, but causing a decline in accuracy (blue). As defined in Sec. 5.3, we will assume that we are interested in EO and DP ratios of at least 0.95. To estimate the corresponding accuracies, we assume linearity between the threshold values we tested for and interpolate accordingly. For EO, we would thus have to pick $t_{A=0} \approx 0.43$ ($1 - t_{A=0} \approx 0.57$) to get an EO ratio of 0.95, resulting in an approximate accuracy of 0.782. Changing the threshold from 0.5 to 0.43 is a relatively small fairness intervention. Consequently, the accuracy of the EO-corrected model is close to the original accuracy (0.790 at $t_{A=0} = 0.5$). For DP, the more drastic measure, we need a drastic fairness intervention, lowering the threshold for being classified attractive to

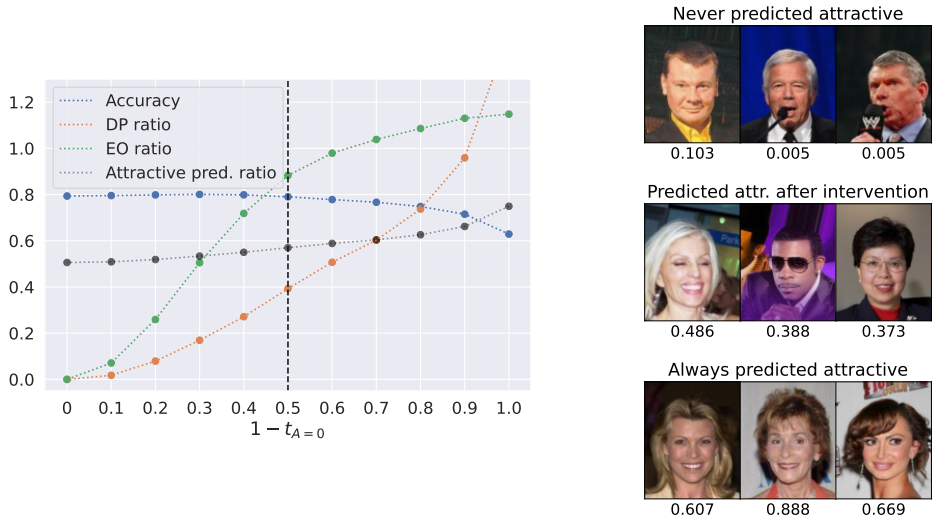


Figure 5.2: *Left:* Accuracy, DP and EO ratio and the total percentage of individuals predicted attractive as a function of the minority group's threshold $t_{A=0}$. We plot $1 - t_{A=0}$ on the x-axis, such that larger values correspond to more drastic fairness interventions. *Right:* Samples and their predictions. The number below each image refers to $P(attractive)$ according to our model. The top row contains images that are never predicted attractive (i.e. $P(attractive) < 0.104$) and the middle row contains images that were predicted attractive after the DP fairness intervention (i.e. $0.104 < P(attractive) < 0.5$, 0.104 is the threshold that produces a DP ratio of 0.95, see main text for discussion). The bottom row is always considered attractive (i.e. $0.104 > P(attractive) > 0.5$).

$t_{A=0} = 0.103$, resulting in an approximate accuracy of 0.7163. These values are compared to the other techniques in Table 5.1.

5.3.2 Naive Upsampling

In Sec. 5.1 we have identified two types of data bias. Firstly, old people are under-represented in the CelebA dataset (22.64%).[2] Secondly, amongst the old demographic group, there are less people labelled attractive (15.42%). From

[2]This is assuming that the label simply refers to whether someone is younger or older than the median age. As noted at the beginning of the chapter, the dataset curators Liu et al. [2015] do not specify the criteria by which the age label was assigned.

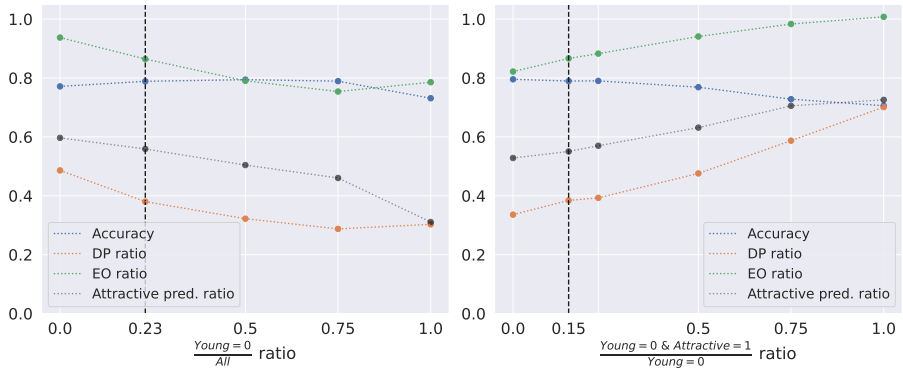


Figure 5.3: *Left:* Upsampling old individuals ($Young = 0$) such that the ratio of old individuals corresponds to the values on the x-axis (we abbreviate these ratios with r in the text). On the y-axis are accuracy, DP and EO ratio as well as the total percentage of individuals predicted attractive. *Right:* Corresponding plot for upsampling of individuals which are labelled both old and attractive. The number on the x-axis is now the ratio of *attractive* old people to old people. The dashed line indicates the original ratios in the unmanipulated dataset.

this, we can derive two re-weighting or upsampling strategies: We can artificially increase the number of old people in general, or of the old people that were labelled attractive.

Figure 5.3 illustrates the effect of those two upsampling strategies. Upsampling old individuals (left panel) and upsampling those labeled both old and attractive (right panel). We might hope that upsampling old individuals would improve EO rates: The model is presented with more old individuals than before, and might thus be making more accurate predictions, including more true positives as measured by EO, on this population. However, this is hardly the case. The EO rate drops with increased oversampling of $A = 0$ individuals (probably because the overall proportion of attractive samples in the training data decreases). Similarly, the best results are achieved when we only sample young individuals ($r = \frac{Young=0}{All}$ ratio $= 0$), and, as a consequence, increase the overall proportion of attractive individuals in the training data. As expected, no improvements in DP are achieved with this upsampling strategy.

The second upsampling strategy, i.e. upsampling old, attractive individuals (Fig. 5.3, right panel) is more successful. As expected, increasing the proportion of attractive individuals amongst the old population improves both

EO and DP ratios. For EO, we would have to pick an upsampling factor of approx. $r = 0.553$ to get an EO ratio of 0.95, resulting in an approximate accuracy of 0.760, a moderate drop from the original accuracy of 0.790. Notably, a DP ratio of 0.95 is not achievable using this technique, the best possible DP ratio is 0.701 at an upsampling factor of 1. See Table 5.1 for direct comparison to other methods.

5.3.3 Data Augmentation for Bias-Correction

We will now investigate whether (learned) data augmentation is a better debiasing tool than naive upsampling. In both cases we present the model with the minority group samples more often than with the majority group ones. In the naive case, samples are simply repeated and the model is thus presented with identical copies. Under the upsampling scheme with data augmentation, the sample is slightly perturbed each time it is fed to the model. In order not to introduce spurious correlations between the demographic group and augmentation, we also augment the majority example whenever those are sampled.

We design the data augmentation scheme by using the Probabilistic Spatial Transformer (P-STN) model from Sec. 2.2.2 and predict 4-parameter affine transformations. Thus, our augmentation transformations are T_ϕ with $\phi = [r, s, t^x, t^y]$, i.e. we learn rotations, scale as well as translations in $x-$ and $y-$direction. We fix the localiser mean to the identity transform and only learn transformation variance β, closely mimicking a standard neural network with data augmentation in the usual sense — however, we marginalise the augmentations as per usual in the P-STN model, see Eq. 4.1. We compare two upsampling schemes as before: 1) upsampling all individuals with $A = 0$ and 2) upsampling those individuals with $A = 0$ and $Y = 1$.

As in the naive upsampling case in Sec. 5.3.2, upsampling all old individuals is not a useful strategy (Fig. 5.4, left panel). Also consistent with the previous experiment, upsampling attractive old individuals improves fairness under both EO and DP. We achieve our target EO ratio of 0.95 at an upsampling rate of $r \approx 0.610$. This is, unexpectedly, a more drastic intervention than was necessary in the naive case (where we found $r \approx 0.553$). The accuracy at this upsampling rate is 0.764, higher than in the naive case. The original accuracy for the P-STN model is 0.801, about one percent point higher than the naive model. See Table 5.1 for a complete comparison. As in the naive case, a DP ratio of 0.95 cannot be achieved using this debiasing method.

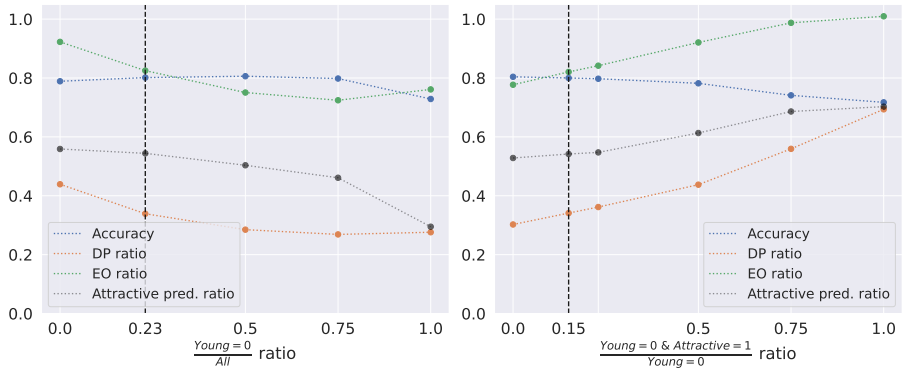


Figure 5.4: Same plot as in Fig. 5.3, but for the DA-upsampling strategy. As before, *left:* upsampling old individuals ($Young = 0$) such that the ratio of old individuals corresponds to the values on the x-axis. On the y-axis are accuracy, DP and EO ratio as well as the total percentage of individuals predicted attractive. *Right:* Upsampling individuals which are labelled both old and attractive. The number on the x-axis is now the ratio of *attractive* old people to old people. The dashed line indicates the original ratios in the unmanipulated dataset.

5.4 Summary

We have evaluated three debiasing techniques: adjusting thresholds, naive upsampling and upsampling using our P-STN model. We have quantified their success by evaluating their fairness-accuracy tradeoff. Specifically, for a given fairness level, i.e. EO and DP ratios of 0.95, we estimate how accurate the methods are. We summarise the results in Table 5.1 below. Recall that upsampling all old individuals did not prove to be a useful strategy. Hence, the results in Table 5.1 for the upsampling methods are all using 'strategy 2', i.e. upsampling attractive old individuals.

We might conclude from Table 5.1 that adjusting thresholds is the most promising upsampling strategy: It is the only method that can achieve the desired DP ratio of 0.95, and achieving the this desired ratio for EO comes at less of a loss in terms of accuracy than for the other methods. While this can certainly be considered successful debiasing, recall that this method requires access to the protected attribute A at test time, which we might not always be given. The upsampling-based methods do not require test-time access to A. Amongst those, DA upsampling achieves better accuracy at the desired EO

Target Metric	Method	Orig. Acc.	t_A / r_A ↓	Acc. at r_A/t_A ↑
EO ratio > 0.95	Thresholding	0.790	$t_A = 0.430$	**0.782**
	Naive Upsamp.	0.790	$r_A = 0.553$	0.760
	DA Upsamp.	**0.801**	$r_A = 0.610$	0.764
DP ratio > 0.95	Thresholding	0.790	$t_A = 0.104$	**0.716**
	Naive Upsamp.	0.790	–	–
	DA Upsamp.	**0.801**	–	–

Table 5.1: Comparison of the different debiasing techniques presented in this chapter for both target metrics (EO in the top half, DP in the bottom half). For each of the three methods, we report the original accuracy, the magnitude of the neccessary fairness intervention (i.e. the adjusted thresholds t_A and sampling rates r_A, respectively), and the accuracy after the fairness intervention. Note that the values in t_A/r_A are not directly comparable between the first and the two last methods, since they can either refer to thresholds or sampling rates.

ratio. On the other hand, DA upsampling requires a more drastic intervention, i.e. the rate at which we need to upsample the attractive old population is higher (0.610 as compared to 0.553 for naive upsampling). Consequently, the relative drop in accuracy is lower for the naive model. In other words, the fact that DA upsampling retains higher accuracy at the desired fairness level might be better explained by the higher original accuracy of the P-STN-like DA model, rather than by improved debiasing behavior.

Which debiasing strategy one should pick depends on the context of the exact problem at hand. Will demographic information be available at test-time, and do we care how drastic the intervention is? Does accuracy retention matter much, or are we building a fair model at any cost? The question at how to arrive at a useful measurement or modelling strategy for algorithmic fairness is the topic of Paper 3, which we will discuss in the next chapter.

Chapter 6

A Closer Look at Fairness Modelling

This chapter summarises the contributions from Paper 3 and applies them to the fairness problem discussed in the last chapter. Thus, the theory is a reproduction while the application is original work of this thesis.

In the previous chapter, we have brushed over the choice of fairness metric, and have without much consideration chosen the well-known fairness metrics demographic parity and equality of opportunity (Def. 5.1) to evaluate our upsampling strategy against. In any real world application, however, the choice of fairness metric is of crucial importance: it encapsulates our understanding of fairness and its operationalisation.

Much efforts by the fairness community have gone into proposing a multitude of fairness metrics. Paper 3 is a critical review of this body of work. We identify two main shortcomings of conventional fairness metrics, give examples for how and why these approaches fail, and then arrive at an alternative modelling strategy which we call *dynamical fairness modelling*. In this chapter, we will summarise the argument from Schwöbel and Remmers [2022] and will then investigate how the proposed dynamical fairness modelling approach can be applied in the context of the CelebA debiasing experiment from Chapter 5.

6.1 Two Shortcomings of the Fair ML Literature

Paper 3 identifies two shortcomings of the established fairness literature. We summarise the argument from Sec. 1 and 2 of the paper below.

Firstly, the ethical and formal debate are often conflated by the fair ML literature, which we are not the first authors to note [Jacobs and Wallach, 2021, Binns, 2020]. As a consequence, the usual fairness analysis is focused too narrowly on mathematical aspects of the metrics alone. For example, consider the tension between the fairness metrics demographic parity and individual fairness: It can be shown that demographic parity and individual fairness contradict each other. We have defined Demographic Parity in Def. 5.1 and will define individual fairness here:

Definition 6.1 (Individual Fairness) Let $x_1,\ x_2 \in \mathcal{X}$ be two inputs to our model, usually thought to represent feature vectors describing two individuals. Let $f : \mathcal{X} \to \mathcal{Y}$ be the model mapping inputs to outputs. D is a metric in the input space and d a metric in the output space. Then f satisfies individual fairness iff

$$d(f(x_1), f(x_2)) \leq D(x_1, x_2). \tag{6.1}$$

Intuitively, if individuals have similar features, they should be assigned similar outcomes. For example, in a hiring scenario, people with similar qualifications should be accepted at the same rates. Importantly in the fairness context, this should hold when both stem from different demographic groups. This notion of fairness is sometimes traced back to Aristotle's principle of 'treating like cases alike' [Schwöbel and Remmers, 2022, Binns, 2020]. Recall that demographic parity poses the requirement to assign the positive outcome (i.e. getting the job in the hiring example) at the same rate to all demographic groups. Comparing the two fairness metrics it is easy to see that they might not be satisfiable at the same time. When the underlying feature distribution is different between demographic groups (say, men are on average more qualified for the job in question) we will hire more men than women if we want to satisfy individual fairness. On the contrary, if we ensure demographic parity and thus hire the same amount of men and women, there will be some men that do not get hired despite being better qualified than their female counterparts which did get the job — breaching individual fairness. This conflict between individual and group fairness (demographic parity specifically) is often discussed on a formal level alone, and is treated as a technical flaw of the metric themselves. However, as Binns [2020] argues

convincingly, the apparent conflict between those two metrics can be easily resolved by taking into account the shared moral principle of egalitarianism, i.e. the idea that '[p]eople should get the same, or be treated the same, or be treated as equals, in some respect.'[1]. As shown in Binns [2020], this ethical stance underlies both group and individual fairness metrics. Grounding the formalised debate in an ethical perspective is helpful — and crucial given the nature of the questions fair ML research seeks to answer.

Secondly, fairness modelling often fails to account for context, specifically the effects of interventions. The second problem we identify with existing fairness metrics is that they often do not sufficiently contextualise the problem at hand. To illustrate this, we briefly summarise the argument from Paper 3, Sec. 2. *Procedural* fairness criteria assume that a fair process is sufficient to ensure a fair state of the world. This notion of fairness is what Rawls [2009] calls pure procedural justice, as opposed to (im-)perfect procedural justice. The latter considers *outcomes* rather than processes when evaluating fairness. For example, demographic parity and equality of opportunity from Def. 5.1 fall into this category: we evaluate the statistical distribution of outcomes in order to arrive at conclusions about our algorithm's fairness. On the procedural side, one of the simplest metrics is Fairness through Unawareness [Gajane and Pechenizkiy, 2017].

Definition 6.2 (Fairness Through Unawareness) A predictor f satisfies fairness through unawareness iff it does not use the protected attribute, i.e.

$$f(X, A) = f(X). \tag{6.2}$$

It is easy to see the limitations of this fairness concept. Many processes in society already satisfy formal equality of opportunity as required here, i.e. demographic information is not usually actively used in decision processes. In fact, in many countries it is illegal to use features like gender, religious affiliation of age to make decisions about individuals — that is why those features are referred to as protected attributes in the literature. In the US, for example, this is implemented via laws such as the US Civil Rights Act Title VII in the context of hiring, or in the US Fair Housing Act. Despite not using protected attributes actively many decision processes produce unfair structures nonetheless. In the infamous COMPAS case [Angwin et al., 2016], an algorithm supposed to help judges by predicting offenders' recidivism risks, the race attribute is not actively used. Yet, the outcomes are massively different for Black and white offenders. In particular, the algorithm errs differently for

[1] From `https://plato.stanford.edu/entries/egalitarianism`.

both demographic groups, producing more false negatives for whites, but false positives for Blacks (i.e. assigning high risk scores despite the person not actually re-offending, the 'undesirable' type of error for the individual). Due to the biased training data, the algorithm learns to correlate race with crime via other factors such as geography or socioeconomics. Similarly in hiring: Not explicitly stating one's gender on a CV is not usually enough to remove any gender information, since such information is encoded in the name, but also in less obvious features such as volunteering or extracurriculars like certain sports which tend to correlate with gender. Thus, the formal exclusion of protected attributes does not stop the protected attribute from (indirectly) impacting decision processes. Algorithms that are learned from historic data are likely to re-enforce existing inequalities. Outcome based metrics such as demographic parity can potentially lead to more in-depth interventions, but this group of metrics usually fails to capture ethically relevant structural differences between groups. They might mark an imbalanced distribution of outcomes as unfair, but they do not help investigate *how* differences in outcome distributions arose historically, or how intervening on them it will change the situation for the underrepresented group in the long term. For example, well-meant affirmative action can sometimes have negative consequences for the disadvantaged group it is supposed to benefit (see Paper 3, Sec. 2.2).

In summary, we have identified two shortcomings of much of the existing fairness literature. Firstly, it often fails to state ethical goals explicitly before starting to develop formalisations. Secondly, it usually fails to evaluate the possible consequences of an intervention by phrasing fairness as a static problem rather than considering societal dynamics over time.

6.2 Dynamical Fairness Modelling

As a consequence, Paper 3 arrives at the following three step approach to fairness modelling (Schwöbel and Remmers [2022], Sec. 3.1).

(1) Explicating ethical goals is the first step in the dynamical fairness modelling pipeline. Importantly, this is done independently of formalisations, i.e. not in mathematical language. Ethical goals can be developed in reference to philosophical stances on fairness or justice, but importantly they should also be developed practically, applied to the context, and ideally in collaboration with different stakeholders in the problem at hand.

(2) Formalisation is the second step: we try to operationalise the ethical goals developed in the first step. This might be as simple as picking a suitable one from the large body of existing fairness metrics, but ethical goals should not be expected to always correlate with existing metrics one-to-one. In particular, as illustrated in detail in Sec. 3.3 and 3.4 of Paper 3, formalisations that can capture the long term effects of fairness interventions might be more appropriate than static, statistical metrics. For example, as proposed in Liu et al. [2019], we might want to explicitly optimise for an *improvement* of the disadvantaged group's living conditions over time.

(3) Modelling down-stream effects of any potential fairness intervention is the third step. The purpose of algorithmic fairness as we see it is to help make positive change in society. In Paper 3, we refer to this as the *interventional perspective* on fairness modelling. Under this perspective, we consider a decision fair if it has the desired, fair outcomes, i.e. if it produces improved conditions for the previously disadvantaged. If a robust estimate of downstream effects is available — we acknowledge that this might be extremely difficult in practice — considering those is a good way to choose one intervention over the other. In the context of gender imbalance in the workplace, for example, we might be able to determine whether women's quotas or more early career initiatives (i.e. fixing the 'pipeline problem') might be the better fairness intervention (see Paper 3, Sec. 3.2).

6.3 Case Study

In Paper 3, we illustrate our dynamical fairness modelling framework using the example of women on company boards. Women are underrepresented in this category of high-impact jobs, and the EU countries have since 2006 taken measures to close this gender gap. Most prominently, they have considered women's quotas as a somewhat controversial measure. Observing a primarily US-centric debate in the fairness space, we apply our dynamical fairness modelling framework to this problem in Sec. 3.2 as a European case study. Here, we will instead use another case-study for illustration, one that matches closely the upsampling experiments from Chapter 5.3.3.

Recall the literature we reviewed at the beginning of Chapter 5.3.3. Authors such as Dion et al. [1972], Shahani-Denning [2003] and Pfeifer [2012] have demonstrated how being perceived attractive has effects on personal outcomes of extreme importance in the personal as well as professional domain: People who are considered attractive are assumed by their peers to make better

friends, partners or employees. Given these positive associated outcomes, a government might decide to try and push forward a more diverse and inclusive beauty standard. In Chapter 5.3.3 we have considered beauty standards with respect to age, but this is equally relevant regarding factors like body image, race or gender expressions. In the following, we will see how the dynamical fairness modelling framework can be applied in this scenario in order to arrive at a more grounded conception of fairness than we have provided previously.

(1) Ethical goals of such an intervention might be diversity, inclusivity and representation in society. From the positive outcomes associated with being considered attractive, we could derive goals such as equal opportunity and social mobility (since people who are considered attractive by their peers have better social and professional outcomes). Lastly, there is a damage control aspect to certain beauty standards: Along the dimension of weight and body image, eating disorder have been on the rise throughout the last 20 years [Galmiche et al., 2019, Morris and Katzman, 2003]. Individuals who suffer from eating disorders face complication ranging from decreased quality of life to physical and psychiatric symptoms, and significantly higher mortality rates [Galmiche et al., 2019]. Researchers have long hypothesised that the media play a strong role in people's body dissatisfaction and may thus be responsible for the increase in eating disorders, especially among young people [Morris and Katzman, 2003]. Thus, there is likely a very direct link between beauty standards as portrayed by the media and public health.

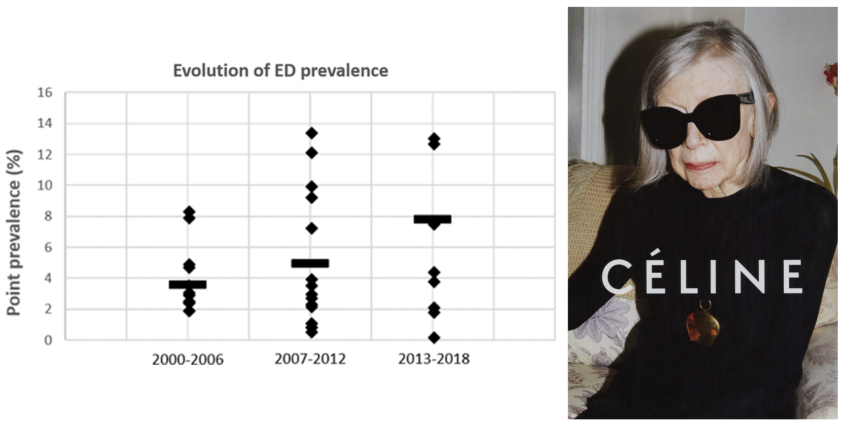


Figure 6.1: *Left:* The prevalence of eating disorders (ED) over time from 2000 to 2018 on a global level. Point prevalence refers to the occurrence at any given point in time. Figure from Galmiche et al. [2019]. *Right:* Age-inclusive advertisement by Céline featuring US American author Joan Didion.

(2) A formalisation of these ethical goals is some version of demographic parity which we have introduced in Def. 5.1. The government might propose a law to require fashion companies to run campaigns which are representative of the underlying demographics (say, on a yearly aggregate level). Our imaginary fashion company from Chapter 5 would have to modify their automatic screening/hiring algorithm such that it achieves this target. In the example in Chapter 3, adjusting thresholds was the most successful strategy for achieving demographic parity. Recall that it required to actively use the protected attribute at test-time in order to be able to determine the correct threshold. This is unacceptable under the fairness through unawareness criteria from Def. 6.2. Given the context of our problem, however, it becomes clear that fairness through unawareness is not the appropriate metric to consider here. Our imaginary government tries to actively use the protected attribute to counter existing problems.

(3) Downstream effects should be considered in the following sense: Demographic parity as a static fairness metric does not quite seem to suffice to express the ethical goals formulated in **(1)**. Our declared moral goal was to change prevailing beauty standards towards being more inclusive and diverse, along with achieving the implied secondary goals with respect to health and economic factors. An ideal formalisation would measure this 'change in beauty standard'. This is of course a non-trivial thing to measure, but one could imagine potential proxies. For example, our fashion house from Ch. 5 could be asked to report the diversity of their pool of applicants for the campaign. The mental model would be that most people will apply only if they feel like they have a real chance of being hired, i.e. if they feel their bodies conform with the prevalent beauty standards. The proxy might be even more reliable if there is a third party acting in the middle, i.e. a modelling agency pre-picking relevant candidates. In either case, if the pool of applicants becomes more diverse over time, this indicates that beauty standards might indeed have changed. Jacobs and Wallach [2021] give an in-depth introduction on how me might apply measurement theory from the social sciences to fair ML in order to make 'unobservable theoretical constructs, such as socioeconomic status, teacher effectiveness, and risk of recidivism' measurable.

6.4 Summary

Picking the right fairness metric is non-trivial, and we have seen that the traditional fair ML literature is not always helpful in this (Sec. 6.1). As a remedy, Paper 3 introduces dynamical fairness modelling. By applying dynamical fairness modelling to the case study from Ch. 5, we have illustrated

how one could pick a useful fairness metric. To do so, we have **1)** developed a set of ethical goals, **2)** defined a formalisation and **3)** derived a strategy for intervention under consideration of its downstream effects.

By re-phrasing the problem in this sense we were able to rule out some candidate metrics: Fairness through unawareness will not produce a change at all (since our model in Ch. 5 is using image features alone, i.e. it is not using demographic attributes explicitly). Applying equality of opportunity will also not result in societal change, since it only measures how well the model re-produces the status quo (how well predictions match ground truth). Lastly, demographic parity turned out to be the most promising candidate, but the dynamical modelling perspective suggests a modification: we might measure the downstream effects of our intervention by measuring whether and how our applicant pool changes over time. If our intervention is successful, the applicant pool will look more diverse (since more candidates consider themselves qualified). Lastly, if societal standards indeed have changed sufficiently, the fairness intervention potentially becomes obsolete all together. As we argue in Paper 3: 'The goal of a good fairness intervention is that it webwill become redundant over time.'

Chapter 7

Final Remarks

The fairness of ML algorithms has, rightfully, been under heavy scrutiny in recent years with cases like COMPAS [Angwin et al., 2016] surfacing. Given the mechanisms by which machine learning works, model bias can very often be traced back to data bias ("*Garbage in — garbage out.*"[1]). Re-collecting better, representative and unbiased datasets would be the ideal solution, or maybe not automating decisions with high moral stakes in the first place. In this work, however, we investigate whether synthetic data can be used to alleviate such data biases and produce fairer models where other options are not available. In particular, we study data augmentation and invariance learning.

Learning data augmentation has turned out to be non-trivial, and have seen that naive likelihood maximisation approaches are insufficient (Ch. 2). We have then re-phrased the problem into a Bayesian model selection problem which we tackled using GP-based methods for approximating and maximising the marginal likelihood. As an alternative approach to achieving model invariance, we have studied pose-normalising models like the spatial transformer network and our probabilistic extension in Ch. 4.

Data augmentation and model invariances have relevance much beyond the fairness application and have been a pursuit of artificial intelligence research since the early days of the field [Pitts and McCulloch, 1947, Minsky, 1961]. As

[1] See `https://en.wikipedia.org/wiki/Garbage_in,_garbage_out`.

we have seen in Chapters 2 to 4, they represent useful inductive biases with remarkable benefits such as improved predictive accuracy, robustness and calibration. While we have not considered such applications in this work, they also play a role for interpretability and when aiming to learn disentangled representations.

Returning to the fairness application, we have conducted a case study on the CelebA [Liu et al., 2015] dataset in Chapter 5. We compare upsampling using data augmentation with other de-biasing strategies such as adjusting group-based thresholds. Thresholding-based methods were in general more easily applicable, but have the big disadvantage that we need to know which threshold to apply at test-time. In non-technical terms, we need to actively use the group membership (e.g. race, age, or gender) in our decision-making process. This is illegal according to anti-discrimination law in many contexts.

Which de-biasing algorithm to apply depends, in the end, on one's measure of fairness. A large body of fairness metric exists, and picking the right one can be tricky. Chapter 6 aims to simplify such choices by grounding the formalised fairness debate in ethical considerations. In the end, questions around fairness and justice are not mathematical in nature, and we believe that the technical community can benefit from looking to fields such as ethics or political philosophy for guidance.

Bibliography

Report: Racial diversity ticks up slightly, size, age and gender representation all drop for fashion month spring 2021, Oct 2020. URL https://www.thefashionspot.com/runway-news/858789-diversity-report-fashion-month-spring-2021/.

The fashion world promised more diversity. here's what we found., Mar 2021. URL https://www.nytimes.com/2021/03/04/style/Black-representation-fashion.html.

J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias. *ProPublica, May*, 23 (2016):139–159, 2016. URL https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

Arch. How to calculate and increase your tinder elo score, July 2020. URL https://social.techjunkie.com/calculate-increase-tinder-elo-score/.

S. Barocas, M. Hardt, and A. Narayanan. Fairness in machine learning. *NIPS tutorial*, 1:2, 2017.

G. Benton, M. Finzi, P. Izmailov, and A. G. Wilson. Learning invariances in neural networks. In *Advances in Neural Information Processing Systems*, 2020.

R. Binns. On the apparent conflict between individual and group fairness. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*, page 514–524. ACM, Jan. 2020. doi: 10.1145/3351095.3372864. URL https://doi.org/10.1145/3351095.3372864.

C. M. Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

T. J. Brinker, A. Hekler, A. H. Enk, C. Berking, S. Haferkamp, A. Hauschild, M. Weichenthal, J. Klode, D. Schadendorf, T. Holland-Letz, et al. Deep neural networks are superior to dermatologists in melanoma image classification. *European Journal of Cancer*, 119:11–17, 2019.

J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Proceedings of the ACM Conference on Fairness, Accountability and Transparency*, pages 77–91. PMLR, 2018.

Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. 2016.

A. Carman. Tinder says it no longer uses a 'desirability' score to rank people, Mar 2019. URL `https://www.theverge.com/2019/3/15/18267772/tinder-elo-score-desirability-algorithm-how-works`.

O. Chapelle, J. Weston, L. Bottou, and V. Vapnik. Vicinal risk minimization. *Advances in neural information processing systems*, 13, 2000.

E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020.

J. Dastin. Amazon scraps secret ai recruiting tool that showed bias against women. `https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G`, 2018. Accessed: 21.03.2022.

E. Daxberger, A. Kristiadi, A. Immer, R. Eschenhagen, M. Bauer, and P. Hennig. Laplace redux-effortless bayesian deep learning. *Advances in Neural Information Processing Systems*, 34, 2021.

N. S. Detlefsen. libcpab. `https://github.com/SkafteNicki/libcpab`, 2018.

N. S. Detlefsen, O. Freifeld, and S. Hauberg. Deep diffeomorphic transformer networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4403–4412, June 2018.

N. S. Detlefsen, M. Jørgensen, and S. Hauberg. Reliable training and estimation of variance networks. In *33rd Conference on Neural Information Processing Systems*, 2019.

K. Dion, E. Berscheid, and E. Walster. What is beautiful is good. *Journal of personality and social psychology*, 24(3):285, 1972.

C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference on - ITCS '12*, page 214–226. ACM Press, 2012. doi: 10.1145/2090236.2090255. URL `https://doi.org/10.1145/2090236.2090255`.

L. Finnveden, Y. Jansson, and T. Lindeberg. Understanding when spatial transformer networks do not support invariance, and what to do about it. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 3427–3434. IEEE, 2021.

P. Gajane and M. Pechenizkiy. On formalizing fairness in prediction with machine learning. *arXiv preprint arXiv:1710.03184*, 2017.

Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

M. Galmiche, P. Déchelotte, G. Lambert, and M. P. Tavolacci. Prevalence of eating disorders over the 2000–2018 period: a systematic literature review. *The American journal of clinical nutrition*, 109(5):1402–1413, 2019.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org, 2017.

F. K. Gustafsson, M. Danelljan, and T. B. Schon. Evaluating scalable bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 318–319, 2020.

M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. *Advances in Neural Information Processing Systems*, 29:3315–3323, 2016.

S. Hauberg, O. Freifeld, A. B. L. Larsen, J. W. Fisher, and L. K. Hansen. Dreaming more data: Class-dependent distributions over diffeomorphisms for learned data augmentation. In *Artificial Intelligence and Statistics*, pages 342–350, 2016.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

J. Hensman, A. G. d. G. Matthews, and Z. Ghahramani. Scalable variational Gaussian process classification. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

A. Z. Jacobs and H. Wallach. Measurement and fairness. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*, page 375–385. ACM, Mar. 2021. doi: 10.1145/3442188.3445901. URL https://doi.org/10.1145/3442188.3445901.

M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

L. H. Kamulegeya, M. Okello, J. M. Bwanika, D. Musinguzi, W. Lubega, D. Rusoke, F. Nassiwa, and A. Börve. Using artificial intelligence on dermatology conditions in uganda: A case for diversity in training data sets for machine learning. *BioRxiv*, page 826057, 2019.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *NIPS*, pages 1097–1105. Curran Associates, Inc., 2012.

F. Küppers, J. Kronenberger, A. Shantia, and A. Haselhoff. Multivariate confidence calibration for object detection. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

F. Küppers, J. Kronenberger, J. Schneider, and A. Haselhoff. Bayesian confidence calibration for epistemic uncertainty modelling. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, July 2021.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

Y. LeCun, L. Jackel, L. Bottou, A. Brunot, C. Cortes, J. Denker, H. Drucker, I. Guyon, U. Muller, E. Sackinger, et al. Comparison of learning algorithms for handwritten digit recognition. In *International conference on artificial neural networks*, volume 60, pages 53–60. Perth, Australia, 1995.

L. T. Liu, S. Dean, E. Rolf, M. Simchowitz, and M. Hardt. Delayed impact of fair machine learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, page 3150–3158. PMLR, International Joint Conferences on Artificial Intelligence Organization, Aug. 2019. doi: 10.24963/ijcai.2019/862. URL https://doi.org/10.24963/ijcai.2019/862.

Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

G. Loosli, S. Canu, and L. Bottou. Training invariant support vector machines using selective sampling. *Large scale kernel machines*, pages 301–320, 2007.

J. Lorraine, P. Vicol, and D. Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *Proceedings of the 23rd Conference on International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

S. Lotfi, P. Izmailov, G. Benton, M. Goldblum, and A. G. Wilson. Bayesian model selection, the marginal likelihood, and generalization. *arXiv preprint arXiv:2202.11678*, 2022.

L. Maaten, M. Chen, S. Tyree, and K. Weinberger. Learning with marginalized corrupted features. In *International Conference on Machine Learning*, pages 410–418. PMLR, 2013.

D. J. MacKay. Bayesian interpolation. *Neural computation*, 4(3):415–447, 1992.

M. Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, 1961.

A. M. Morris and D. K. Katzman. The impact of the media on eating disorders in children and adolescents. *Paediatrics & child health*, 8(5):287–289, 2003.

S. U. Noble. *Algorithms of Oppression*. NYU Press, Feb. 2018. ISBN 9781479833641, 1479833649, 9781479849949. doi: 10.2307/j.ctt1pwt9w5. URL `https://doi.org/10.2307/j.ctt1pwt9w5`.

N. Norori, Q. Hu, F. M. Aellen, F. D. Faraci, and A. Tzovara. Addressing bias in big data and ai for health care: A call for open science. *Patterns*, 2(10):100347, 2021.

S. W. Ober, C. E. Rasmussen, and M. van der Wilk. The promises and pitfalls of deep kernel learning. In *Proceedings of the 37th Conference on Uncertainty in Artifical Intelligence (UAI)*, 2021.

C. Pfeifer. Physical attractiveness, employment and earnings. *Applied Economics Letters*, 19(6):505–510, 2012.

W. Pitts and W. S. McCulloch. How we know universals the perception of auditory and visual forms. *The Bulletin of mathematical biophysics*, 9(3):127–147, 1947.

C. E. Rasmussen and Z. Ghahramani. Occam's razor. *Advances in Neural Information Processing Systems*, 2001.

J. Rawls. *A Theory of Justice*. Harvard University Press, July 2009. ISBN 9780674042582, 9780674000773. doi: 10.2307/j.ctvkjb25m. URL `https://doi.org/10.2307/j.ctvkjb25m`.

P. Schwöbel and P. Remmers. The long arc of fairness: Formalisations and ethical discourse. *arXiv preprint arXiv:2203.06038*, 2022.

P. Schwöbel, F. Warburg, M. Jørgensen, K. H. Madsen, and S. Hauberg. Probabilistic spatial transformers for Bayesian data augmentation. *arXiv preprint arXiv:2004.03637*, 2020.

P. Schwöbel, M. Jørgensen, S. W. Ober, and M. van der Wilk. Last layer marginal likelihood for invariance learning. *arXiv preprint arXiv:2106.07512*, 2021.

P. Schwöbel, M. Jørgensen, S. W. Ober, and M. Van Der Wilk. Last layer marginal likelihood for invariance learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3542–3555. PMLR, 2022.

C. Shahani-Denning. Physical attractiveness bias in hiring: What is beautiful is good. *Hofstra Horizon*, pages 14–17, 2003.

P. Simard, B. Victorri, Y. LeCun, and J. S. Denker. Tangent prop – a formalism for specifying selected invariances in an adaptive network. In *NIPS*, pages 895–903, 1992.

P. Y. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *2013 12th International Conference on Document Analysis and Recognition*, volume 2, pages 958–958. IEEE Computer Society, 2003.

A. Stirn and D. A. Knowles. Variational variance: Simple and reliable predictive variance parameterization. *arXiv e-prints*, pages arXiv–2006, 2020.

H. Takahashi, T. Iwata, Y. Yamanaka, M. Yamada, and S. Yagi. Student-t variational autoencoder for robust density estimation. In *IJCAI*, pages 2696–2702, 2018.

M. Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.

R. E. Turner and M. Sahani. Two problems with variational expectation maximisation for time-series models. In D. Barber, T. Cemgil, and S. Chiappa, editors, *Bayesian Time Series Models*, chapter 5, pages 109–130. Cambridge University Press, 2011.

M. van der Wilk, M. Bauer, S. T. John, and J. Hensman. Learning invariances using the marginal likelihood. In *Advances in Neural Information Processing Systems*, 2018.

B. S. Veeling, J. Linmans, J. Winkens, T. Cohen, and M. Welling. Rotation equivariant CNNs for digital pathology. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018.

C. K. I. Williams and C. E. Rasmussen. *Gaussian processes for machine learning*. MIT Press Cambridge, MA, 2006.

A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016a.

A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Stochastic variational deep kernel learning. *Advances in Neural Information Processing Systems*, 2016b.

J. T. Wilson, V. Borovitskiy, A. Terenin, P. Mostowsky, and M. P. Deisenroth. Efficiently sampling functions from Gaussian process posteriors. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

M. B. Zafar, I. Valera, M. Gomez-Rodriguez, and K. P. Gummadi. Fairness constraints: A flexible approach for fair classification. *The Journal of Machine Learning Research*, 20(1):2737–2778, 2019.

Paper A

Last Layer Marginal Likelihood for Invariance Learning

Last Layer Marginal Likelihood for Invariance Learning

Pola Schwöbel
Technical University of Denmark

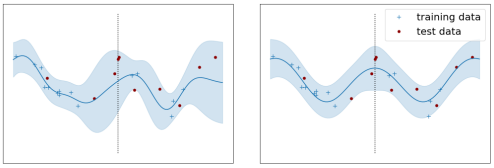
Martin Jørgensen
University of Oxford

Sebastian W. Ober
University of Cambridge

Mark van der Wilk
Imperial College London

Abstract

Data augmentation is often used to incorporate inductive biases into models. Traditionally, these are hand-crafted and tuned with cross validation. The Bayesian paradigm for model selection provides a path towards end-to-end learning of invariances using only the training data, by optimising the marginal likelihood. Computing the marginal likelihood is hard for neural networks, but success with tractable approaches that compute the marginal likelihood for the last layer only raises the question of whether this convenient approach might be employed for learning invariances. We show partial success on standard benchmarks, in the low-data regime and on a medical imaging dataset by designing a custom optimisation routine. Introducing a new lower bound to the marginal likelihood allows us to perform inference for a larger class of likelihood functions than before. On the other hand, we demonstrate failure modes on the CIFAR10 dataset, where the last layer approximation is not sufficient due to the increased complexity of our neural network. Our results indicate that once more sophisticated approximations become available the marginal likelihood is a promising approach for invariance learning in neural networks.



(a) M1:
Train MSE: **0.047**
Test MSE 0.572
Marg. Likelihood −13.16

(b) M2:
Train MSE 0.100
Test MSE **0.235**
Marg. Likelihood **−12.2**

Figure 1: A non-invariant model M1 and its sign invariant (i.e. symmetric around $x = 0$) counterpart M2. The non-invariant M1 has a better train MSE, but the invariant M2 has a better test MSE. The log marginal likelihood correctly identifies M2 as better.

1 INTRODUCTION

Human learners generalise from example to category with seemingly little effort. Machine learning models aim to make accurate predictions on unseen data points based on finitely many examples. This generalisation is enabled by *inductive biases*. In *Steps toward Artificial Intelligence* Marvin Minsky (1961) highlights the importance of invariance as an inductive bias: '*One of the prime requirements of a good property is that it be invariant under the commonly encountered equivalence transformations. Thus for visual Pattern-Recognition we would usually want the object identification to be independent of uniform changes in size and position.*' In modern machine learning pipelines invariances are achieved through data augmentation. If we, for example, would like our neural network to be invariant with respect to rotation, we simply present it with rotated versions of the input data. Data augmentation schemes are almost always hand-crafted, based on assumptions and expert knowledge about the data, or found by cross-validation. We aim to learn invariances with backpropagation, to reduce the human intervention in the design of ML algorithms.

Learning invariances through gradients requires a suit-

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

able loss function. Standard losses like negative log-likelihood or mean squared error solely measure how tightly we fit the training data. Good inductive biases (e.g. convolutions) constrain the expressiveness of a model, and therefore do not improve the fit on the training data. Thus, they can not be learned by minimising the training loss alone.

In Bayesian inference, this problem is known as *model selection*, and is commonly solved by using a different training objective: the marginal likelihood. For a model of data y, parametrised by weights w and hyperparameters θ it is given by

$$p(y|\theta) = \int p(y|w)p(w|\theta)\mathrm{d}w\,. \tag{1}$$

As opposed to standard training losses, it correlates with generalisation, and thus provides a general way to select an inductive bias, independent of parameterisation (Williams and Rasmussen, 2006; Rasmussen and Ghahramani, 2001; MacKay, 2003). Van der Wilk et al. (2018) demonstrated that invariances can be learned by straightforward backpropagation using the marginal likelihood in Gaussian process (GP) models, where the marginal likelihood can be accurately approximated. Fig. 1 shows an invariant and a non-invariant GP; the invariant model has higher marginal likelihood as well as lower test mean squared error. Thus, the marginal likelihood correctly identifies invariance as a useful inductive bias.

Current GP models often lack predictive performance compared to their highly expressive neural network counterparts, hence applying this elegant principle to neural networks is attractive. The challenge is, however, that finding accurate and differentiable marginal likelihood approximations for neural networks is still an open problem. In this work we investigate a convenient short-cut: computing Bayesian quantities only in the last layer. This avoids difficulties of the marginal likelihood in the full network, and has already been shown helpful (Wilson et al., 2016a,b). Given the possible impact of invariance learning with the convenience of the last-layer approximation, it is important to investigate its potential. Our results provide a nuanced picture of this approach: there are situations where the last-layer approximation is sufficient, but others where it is not.

To provide these results, we

1. construct a *deep neural network with a Bayesian last layer that incorporates invariance*, based on invariant GPs (van der Wilk et al., 2018) and deep kernel learning (Wilson et al., 2016b),

2. overcome problems with the training implied by a straightforward combination of Van der Wilk et al. (2018) and Wilson et al. (2016b) via a *new optimisation scheme*, and *a new variational bound* that allows for non-Gaussian likelihoods,

3. *investigate failure modes* on more complex model architectures to show limitations of using the last-layer approximation for invariance learning.

2 RELATED WORK

Bayesian Deep Learning aims to provide principled uncertainty quantification for deep models. Exact computation for Bayesian deep models is intractable, so different approximations have been suggested. Variational strategies (e.g. Blundell et al., 2015) maximise the evidence lower bound (ELBO) to the marginal likelihood, thereby minimising the gap between approximate and true posteriors. To remain computationally feasible, approximations for Bayesian neural networks are often crude, and while weight posteriors are useful in practice, the marginal likelihood estimates are typically too imprecise for hyperparameter estimation (Blundell et al., 2015; Turner and Sahani, 2011). Hyperparameter estimation in deep GPs has achieved more success (Damianou and Lawrence, 2013; Dutordoir et al., 2020), but training deep GPs can be challenging. Some very recent works have shown initial promise in using the marginal likelihood for hyperparameter selection in Bayesian neural networks (Ober and Aitchison, 2020; Immer et al., 2021; Dutordoir et al., 2021). Instead of a Bayesian treatment of all weights using rough approximations, we follow a deep kernel learning approach, i.e. computing the marginal likelihood for the last layer only.

Deep Kernel Learning (DKL; Hinton and Salakhutdinov, 2007; Calandra et al., 2016; Bradshaw et al., 2017) replaces the last layer of a neural network with a GP, where marginal likelihood estimation is accurate (Burt et al., 2020). Wilson et al. (2016a,b) had significant success achieving improved uncertainty estimates. Their results indicate that such a neural network-GP hybrid is promising for invariance learning. Ober et al. (2021) identify difficulties with overfitting in DKL models, but also show mechanisms by which such overfitting is mitigated. We find similar issues and adapt the standard DKL training procedure to avoid them when learning invariance hyperparameters. We will discuss these issues in more depth as we describe our training procedure in Sec. 5.

Data Augmentation is used to incorporate invariances into deep learning models. Where good invariance assumptions are available a priori (e.g. for natural images) this improves generalisation performance and is ubiquitous in deep learning pipelines. Instead of relying on assumptions and hand-crafting, recent approaches

learn data augmentation schemes. Cubuk et al. (2019, 2020) and Ho et al. (2019) train on the validation data, and use reinforcement learning and evolutionary search respectively to find parameters. Zhou et al. (2021); Lorraine et al. (2020) compute losses on validation sets for learning invariance parameters, and estimate gradients w.r.t. them in outer loops. Similar to our work, Benton et al. (2020) learn data augmentations on training data end-to-end, by adding a regularisation term to the negative log-likelihood loss that encourages invariance. They argue that tuning this regularisation term via cross-validation can be avoided, since the loss function is relatively flat. Yet, the method relies on explicit regularisation, and thus on an understanding of the parameters in question. Our method is based on a Bayesian view of data augmentation as incorporating an invariance on the functions in the prior distribution (van der Wilk et al., 2018; Nabarro et al., 2021). This allows the marginal likelihood to be used as an objective for learning invariances. This has many advantages, such as allowing backpropagation from training data, automatic and principled regularisation, and parameterisation independence (see Sec. 5). This makes the marginal likelihood objective a promising avenue for future work, which may want to incorporate invariances whose parameterisations are non-interpretable.

3 BACKGROUND

3.1 Variational Gaussian processes

A Gaussian process (GP) (Williams and Rasmussen, 2006) is a distribution on functions with the property that any vector of function values $\boldsymbol{f} = (f(x_1), \ldots, f(x_N))$ is Gaussian distributed. We assume zero mean functions and real valued vector inputs.

Inference in GP models with general likelihoods and big datasets can be done with variational approximations (Titsias, 2009; Hensman et al., 2015). The approximate posterior is constructed by conditioning the prior on M *inducing variables* $\boldsymbol{u} \in \mathbb{R}^M$, and specifying their marginal distribution with $q(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{m}, \boldsymbol{S})$ (for overviews see Bui et al. 2017; van der Wilk et al. 2020). This results in a variational predictive distribution:

$$q(f(x^*)) = \mathcal{N}\big(\boldsymbol{\alpha}(x^*)^\top \boldsymbol{m}, \quad k(x^*, x^*) - \boldsymbol{\alpha}(x^*)^\top \left(\boldsymbol{K}_{\boldsymbol{zz}} - \boldsymbol{S}\right) \boldsymbol{\alpha}(x^*)\big), \tag{2}$$

where $\boldsymbol{z} \in \mathbb{R}^{M \times d}$ are inducing *inputs*, $\boldsymbol{K}_{\boldsymbol{zz}}$ is the matrix with entries $k(z_i, z_j)$, $\boldsymbol{\alpha}(x^*) = \boldsymbol{K}_{\boldsymbol{zz}}^{-1} k(\boldsymbol{z}, x^*)$, and k is the chosen covariance function.

Variational inference (VI) selects an approximation by minimising the KL divergence of the approximation to the true posterior with respect to the variational parameters $\boldsymbol{z}, \boldsymbol{m}, \boldsymbol{S}$. This is done by maximising a lower bound to the marginal likelihood (the "evidence"), which has the KL divergence as its gap (Matthews et al., 2016). The resulting evidence lower bound (ELBO) is

$$\log p(y) \geq \mathcal{L} = \sum_{n=1}^{N} \mathbb{E}_{q(f(x_n))} \left[\log p(y_n | f(x_n))\right] - \mathrm{KL}[q(\boldsymbol{u}) || p(\boldsymbol{u})]. \tag{3}$$

In exact GPs, (kernel) hyperparameters are found by maximising the log marginal likelihood $\log p(y)$ (Williams and Rasmussen, 2006). For our models of interest, the exact marginal likelihood is intractable. We use the ELBO as a surrogate. This results in an approximate inference procedure that maximises the ELBO with respect to both the variational parameters and the hyperparameters. Optimising the variational parameters improves the quality of the posterior approximation, and tightens the bound to the marginal likelihood. Optimising the hyperparameters hopefully improves the model, but the slack in the ELBO can lead to worse hyperparameter selection (Turner and Sahani, 2011).

3.2 Invariant Gaussian Processes

A function $f : \mathcal{X} \to \mathcal{Y}$ is *invariant* to a transformation $t : \mathcal{X} \to \mathcal{X}$ if $f(x) = f(t(x))$, $\forall x \in \mathcal{X}$, and $\forall t \in \mathcal{T}$. I.e., an invariant function will have the same output for a certain range of transformed inputs known as the *orbit*. A straightforward way to construct invariant functions is to simply average a function over the orbit (Kondor, 2008; Ginsbourger et al., 2012, 2013). We consider a similar construction where we average a function over a data augmentation distribution, which results in an approximately invariant function where $f(x) \approx f(t(x))$ (van der Wilk et al., 2018; Dao et al., 2019). Augmented data samples x_a are obtained by applying random transformations t to an input, $x_a = t(x)$, leading to the distribution $p(x_a|x)$. That is, an approximately invariant function f can be constructed from any non-invariant g as

$$f(x) = \sum_{t \in \mathcal{T}} g(t(x)), \text{ or } f(x) = \int g(x_a) p(x_a|x) \mathrm{d}x_a. \tag{4}$$

Van der Wilk et al. (2018) exploit this construction to build a GP with continuously adjustable invariances. They place a GP prior on $g \sim \mathcal{GP}\left(0, k_g(\cdot, \cdot)\right)$, and since Gaussians are closed under summations, f is a GP too. By construction f is invariant to the augmentation distribution $p(x_a|\cdot)$ and its kernel is given by

$$k_f(x, x') = \iint k_g(x_a, x'_a) p(x_a|x) p(x'_a|x') \mathrm{d}x_a \mathrm{d}x'_a. \tag{5}$$

Non-trivial $p(x_a|x)$ densities present a problem for standard VI, as the kernel evaluations in eq. 2 become

intractable. This is solved by making the inducing variables observations of g rather than the usual f. This ensures that $\boldsymbol{K_{zz}}$ is tractable, as it only requires evaluations of k_g, which makes the KL divergence tractable. When the likelihood is Gaussian, it additionally provides a way to tackle the expected log likelihood:

$$\mathbb{E}_{q(f(x))} \log \mathcal{N}(y; f(x), \sigma^2) = \text{const} - \frac{(y_n - \mu)^2 + \tau}{2\sigma^2} \quad (6)$$

where μ, τ are the mean and variance in (2). Only unbiased estimates of μ, μ^2 and τ are needed for an unbiased estimate of the ELBO. These can be obtained from simple Monte Carlo estimates of k_f (5), and $k(\boldsymbol{z}, x)$.[1]

3.3 Parameterising learnable invariances

The invariance of the GP in (5) is learned by adjusting the augmentation distribution. We parameterise the distribution and treat its parameters as kernel hyperparameters. We learn these by maximising the ELBO. As done in similar work (Benton et al., 2020; van der Wilk et al., 2018), we consider affine transformations. Our affine transformations are controlled by $\phi = (\alpha, s^x, s^y, p^x, p^y, t^x, t^y)$, which describes rotation, scale, shearing and horizontal and vertical translation. We parameterise a family of augmentation distributions by specifying uniform ranges with $\phi_{\min}, \phi_{\max} \in \mathbb{R}^7$ that are to be applied to the input image. Different ranges that are learned on $\phi_{\min}, \phi_{\max}$ correspond to different invariances in $f(\cdot)$. For example, learning $\phi_{\min/\max} = (\pm\pi, 0, 0, 0, 0, 0, 0)$ corresponds to full rotational invariance (sampling any angle between $-\pi$ and π) but no scaling, shearing or translations.

We sample from the resulting $p(x_a|x, \phi_{\max}, \phi_{\min})$ (we will write $p(x_a|x, \phi)$ for brevity) by **1)** sampling the parameters for a transformation from a uniform distribution, **2)** generating a transformed coordinate grid, and **3)** interpolating[2] the image x:

$$x_a = t_\nu(x)\,, \qquad \nu \sim U(-\phi_{\min}, \phi_{\max})\,. \quad (7)$$

Since transforming $t_\nu(x)$ is differentiable, this procedure is reparameterisable w.r.t. $\phi_{\max}, \phi_{\min}$ via $\nu = \phi_{\min} + (\phi_{\max} - \phi_{\min})\varepsilon$, $\varepsilon \sim U(0, 1)$. Straightforward automatic differentiation of the unbiased ELBO estimator described in the previous section provides the required gradients.

In summary, we learn $\phi_{\min/\max}$ by maximising the ELBO, so the transformations and their magnitudes are learned based on the specific training set. Different invariances will be learned for different training data. The next sections show how these principles have potential even in neural network models, beyond the single layer GPs of Van der Wilk et al. (2018).

[1] We obtain $k(\boldsymbol{z}, x) = \int k_g(\boldsymbol{z}, x_a) p(x_a|x) \mathrm{d}x_a$ from the interdomain trick, which can be estimated with Monte Carlo. See Van der Wilk et al. (2018) for details.

[2] Image transformation code from `github.com/kevinzakka/spatial-transformer-network`

Algorithm 1: InvDKGP forward pass

1. Draw S samples from the augmentation distribution $x_a^i \sim p(x_a|x, \phi),\ i = 1...S$.
2. Pass the x_a^i through the neural net h_w.
3. Map extracted features using the non-inv. g.
4. Aggregate samples to obtain inv. $f(x)$ by
   (i) using the unbiased estimators from Sec. 3.2 in the Gaussian case, or,
   (ii) averaging predictions $g(h_w(x_a^i)),\ i = 1, ..., S$ directly in the Softmax case, see (16).

4 MODEL

As discussed in Sec. 1, we aim to learn neural network (NN) invariances through backpropagation, in the same way as is possible for single-layer GPs. Since finding high-quality approximations to the marginal likelihood of a NN is an ongoing research problem, we investigate whether a simpler *deep kernel* approach is sufficient. This uses a GP as the last layer of a NN, and takes advantage of accurate marginal likelihood approximations for the GP last layer. Success with such a simple method would significantly help automatic adaptation of data augmentation in neural network models. We hypothesise that the last layer approximation is sufficient, since data augmentation influences predictions only in the last layer (in the sense that one can construct an invariant function f from an arbitrary non-invariant g by summing in the last layer, eq. 4). See Fig. 2 for a graphical representation and Algorithm 1 for forward pass computations.

Deep Kernels take advantage of covariance functions being closed under transformations of their input. That is, if $k_g(\cdot, \cdot)$ is a covariance function on $\mathbb{R}^D \times \mathbb{R}^D$, then $k_g(h_w(\cdot), h_w(\cdot))$ is a covariance function on $\mathbb{R}^d \times \mathbb{R}^d$ for mappings $h_w : \mathbb{R}^d \to \mathbb{R}^D$. In our case, h_w is a NN parametrised by weights w, and hence w are viewed as hyperparameters of the kernel. The GP prior becomes

$$p(g) = \mathcal{GP}\left(0, k_g(h_w(\cdot), h_w(\cdot))\right). \quad (8)$$

The idea is to learn w along with the kernel hyperparameters. Importantly, this model remains a GP and so the inference described in Sec. 3 applies.

Our invariant model combines the flexibility of a NN $h_w(\cdot)$ with a GP g in the last layer, while ensuring

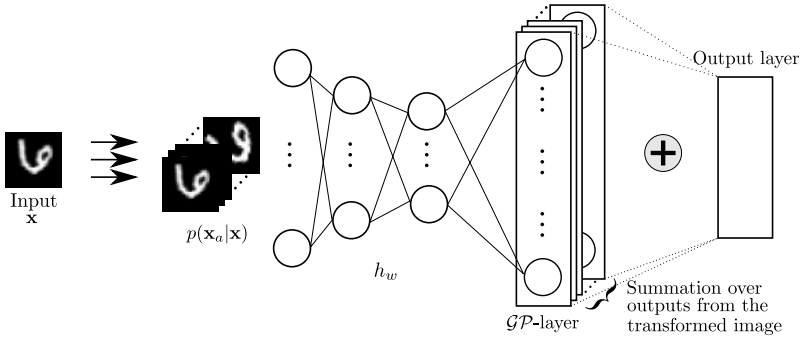


Figure 2: A visualisation of the model pipeline. For any input x, we can sample from the orbit distribution $p(x_a|x, \phi)$; each of these sample gets passed through a neural network parametrised by w. The last layer is a of the net is a GP, on which we can sum across sample outputs to create an invariant function.

overall invariance using the construction from (4):

$$f(x) = \int g(h_w(x_a))p(x_a|x, \phi)\mathrm{d}x_a. \tag{9}$$

Thus, combining (5) and (9), f is an *invariant* GP with a *deep* kernel given as

$$k_f(x, x') = \int k_g\big(h_w(x_a), h_w(x'_a)\big) \\ p(x_a|x, \phi)p(x'_a|x', \phi)\mathrm{d}x_a\mathrm{d}x'_a. \tag{10}$$

The model is trained to fit observations y through the likelihood function $p\,(y|f(x))$, where we assume observations y_i are independent conditioned on the marginals $f(x_i)$.

Initially, we investigate training a model by simply combining the invariant GP training objective for Gaussian likelihoods (van der Wilk et al., 2018) with standard deep kernel learning (Wilson et al., 2016a,b). However, as we will discuss, several issues prevent these training procedures from working. In following sections we investigate why, provide solutions, and introduce a new ELBO that is suitable for more general likelihoods which improves training behaviour. We refer to our model as the *Invariant Deep Kernel GP (InvDKGP)*. An implementation can be found at `https://github.com/polaschwoebel/InvDKGP`.

5 DESIGNING A TRAINING SCHEME

The promise of deep kernel learning as presented by Wilson et al. (2016a,b) lies in training the NN and GP hyperparameters *jointly*, using the marginal likelihood as for standard GPs.[3] However, prior works have noted shortcomings of this approach (Ober et al., 2021; Bradshaw et al., 2017; van Amersfoort et al., 2021): the DKL marginal likelihood correctly penalises complexity for the last layer only, while the NN hyperparameters can still overfit. In our setting, i.e. when trying to combine deep kernel learning with invariance learning, joint training produces overfit weights which results in simplistic features with little intra-class variation[4]. In particular, all training points from the same class are mapped to very similar activations, independent of orientation. This causes a loss of signal for the invariance parameters (see Fig. 3).

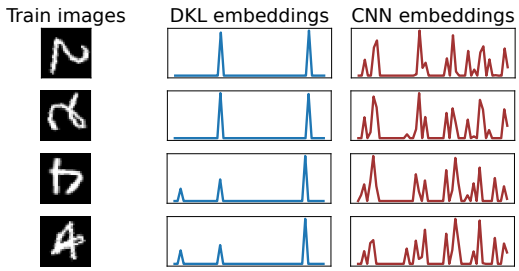


Figure 3: Training images with different orientations and their embeddings. Embeddings produced by joint Deep Kernel Learning (DKL, middle column) are similar for all inputs from one class. Little improvement can be gained on the training data by being rotationally invariant. NN embeddings on the right differ depending on input orientation – signal to learn $p(x_a|x, \phi)$ from.

[3]Given that this quantity is difficult to approximate, we verify experimentally that we indeed need it and cannot use a simple NN with max-likelihood (see Appendix).

[4]This behavior makes sense: The DKL marginal likelihood only penalises complexity in the last layer, (i.e. the GP). The simplistic features from Fig. 3 can be classified by a simple function in the last layer, thus the complexity penalty is small, and the solution has high marg. likelihood.

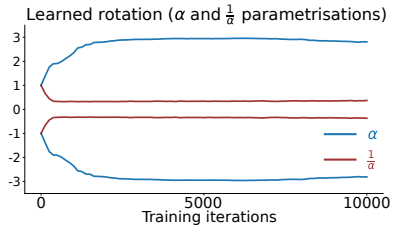


Figure 4: Learned rot. angles parametrised by α and $\frac{1}{\alpha}$. The α-parametrisation, in blue, learns rotational invariance w.r.t. ± 2.8 radians. The $\frac{1}{\alpha}$-parametrisation (red) learns invariance w.r.t. $\pm\frac{1}{0.37} = \pm 2.7$ radians.

Coordinate ascent training fixes this problem. We pre-train the NN using negative log-likelihood loss. Then, we replace the fully connected last layer with an invariant GP. The marginal likelihood is a good objective given fixed weights (we obtain a GP on transformed inputs), so we fix the NN weights. However, *some* adaptation of the NN to the transformed inputs is beneficial. We thus continue training by alternating between updating the NN, and the GP variational parameters and orbit parameters, hereby successfully learning invariances. (See Fig. 7 and 8: flat parts of the training curves indicate NN training where all kernel hyperparameters, including invariances, remain fixed. When to toggle between the GP and NN training phase is determined using validation data.)

Choosing an invariance parameterisation is simple with our method. Other invariance learning approaches, e.g. Benton et al. (2020) and Schwöbel et al. (2020) rely on explicitly regularising augmentation parameters to be large, and thus require interpretability of their parameters. The marginal likelihood objective is *independent of parameterisation*. To illustrate this we compare parameterising the range of angles by the angle in radians α and by its reciprocal $\xi = \frac{1}{\alpha}$. In the rotMNIST example (see Fig 4) large invariances are needed. This corresponds to large α or small ξ – our method obtains this in both parameterisations. In

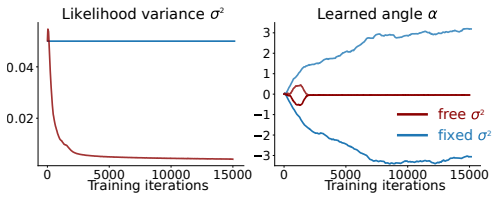


Figure 5: Runs with fixed (red) and non-fixed (blue) kernel and likelihood variance on rotMNIST. The augmentation distribution collapses for non-fixed variances.

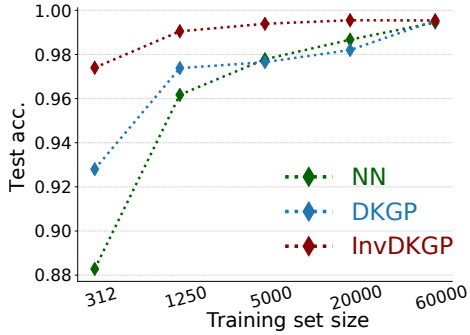


Figure 6: Test accuracies against the training set size on MNIST. We see the invariant model (in red) generalises significantly better, especially for small training sets.

contrast, explicitly regularising invariance parameters to be large would fail for ξ. We wish to stress that generating the orbit distributions is not restricted to affine image transformation and parameterisation independence will be more important as more complicated, non-interpretable invariances are considered.

The Gaussian likelihood is chosen by Van der Wilk et al. (2018) due to its closed-form ELBO. For classification problems, this is a model misspecification. The penalty for not fitting the correct label value becomes large and we can therefore overfit the training data. To alleviate this problem, we fix likelihood and kernel variance (see Fig. 5). The fixed values were determined by trying out a handful candidates – this was sufficient to make invariance learning work. To remove this manual tuning, we will derive an ELBO that works with likelihoods like Softmax in Sec 6.

5.1 MNIST subsets – the low data regime

Having developed a successful training scheme we evaluate it on MNIST subsets. The generalisation problem is particularly difficult when training data is scarce. Inductive biases are especially important and usually parameter-rich neural networks rely on heavy data augmentation when applied to smaller datasets. We train on different subsets of MNIST (LeCun et al.). InvDKGPs outperform both NNs and non-invariant deep kernel GPs. The margin is larger the smaller the training set — with only 1250 training examples we can nearly match the performance of a NN trained on full MNIST (Fig. 6). We conclude it is possible to learn useful invariances even from small data (see Fig. 7). This data efficiency is desirable since models trained on small datasets benefit crucially from augmentation.

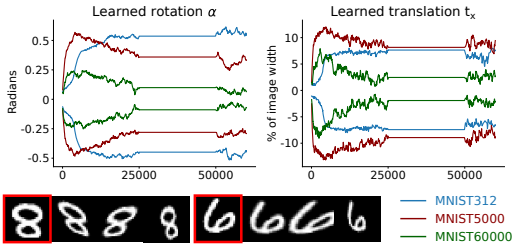


Figure 7: *Top:* Learned invariance parameters (rotation α in radians and x-translation t_x) for a small, medium and large training set. We learn larger α for the smaller subsets. Here, data augmentation is more beneficial. *Bottom*: Two training images x (red frames) and samples from $p(x_a|x, \phi)$ (following columns) learned by the InvDKGP on MNIST using only 312 images.

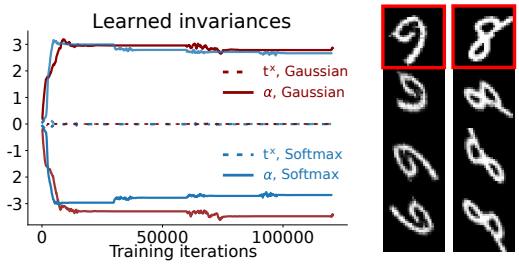


Figure 8: *Left*: Learned invariance parameters (rotation α in radians and x-translation t_x) for rotMNIST. Both the Gaussian and the Softmax model learn to be almost fully rotationally invariant (i.e. $\alpha_{\text{min/max}} \approx \pm\pi$), and not to be invariant w.r.t. translation (i.e. $t^x_{\text{min/max}} \approx 0$). Note the different scaling of the y-axis to Fig. 7. *Right*: Two training images (red frames) and samples from orbits.

6 CORRECTING MODEL MISSPECIFICATION

The key observation for inference under the Gaussian likelihood was the *unbiasedness* of the estimators. In this section, we introduce a controlled bias to allow for easy inference in a wide class of likelihoods. In the limit of infinite sampling, the bias disappears and the invariance does not add additional approximation error.

Recall that $f(x)$ constructed in (4) is intractable but can be estimated by Monte Carlo sampling

$$\hat{f}(x) := \frac{1}{S_o} \sum_{i=1}^{S_o} g(x_a^i), \tag{11}$$

where $x_a^i \sim p(x_a|x, \phi)$. Notice,

$$f(x) = \mathbb{E}_{\prod_{i=1}^{S_o} p(x_a^i|x,\phi)} \left[\hat{f}(x)\right] =: \tilde{\mathbb{E}}\left[\hat{f}(x)\right], \tag{12}$$

where $\prod_{i=1}^{S_o} p(x_a^i|x, \phi)$ is the product density over S_o orbit densities. We remark that f is deterministic in x but stochastic in g, which is a GP. Thus, we can write

$$\mathbb{E}_{q(f(x))}[\log p(y|f(x))] = \mathbb{E}_{q(g)}[\log p(y|f(x))] \tag{13}$$

$$= \mathbb{E}_{q(g)}\left[\log p\left(y|\tilde{\mathbb{E}}[\hat{f}(x)]\right)\right] \tag{14}$$

$$\geq \mathbb{E}_{q(g)}\left[\tilde{\mathbb{E}}\left[\log p\left(y|\hat{f}(x)\right)\right]\right]. \tag{15}$$

The inequality is due to Jensen's inequality if the likelihood is *log-concave* in f.[5] This holds for many common likelihoods, e.g. Gaussian and Softmax.

[5]Nabarro et al. (2021) use this same construction in the weight-space of neural networks to find valid posteriors in the presence of data augmentation, although without invariance learning.

Equality holds above when $\text{Var}(\hat{f}(x)) = 0$, i.e. the bound becomes tighter as S_o increases (see also Burda et al., 2016). Hence aggressive sampling recovers accurate VI. The right-hand side of (15) can now, without additional bias, be estimated by

$$\frac{1}{S_g} \sum_{k=1}^{S_g} \frac{1}{S_\mathcal{A}} \sum_{j=1}^{S_\mathcal{A}} \log p\left(y \middle| \frac{1}{S_o} \sum_{i=1}^{S_o} g_k(x_a^{ji})\right). \tag{16}$$

Since extensive sampling is required to keep the bound above tight, it is important to do this efficiently. From a GP perspective this is handled with little effort by sampling the approximate posteriors $q(g)$ using Matheron's rule (Wilson et al., 2020). Thus, sampling S_g GPs is cheap compared to sampling from the orbit. $S_\mathcal{A}$ denotes the number of $\hat{f}$ samples, this can be fixed to 1 as long as S_o is large.

Summarising, we have shown how we can infer through the marginal likelihood, for the wide class of log-concave likelihoods, by maximising the stochastic ELBO:

$$\mathcal{L} = \frac{1}{S_g} \sum_{k=1}^{S_g} \frac{1}{S_\mathcal{A}} \sum_{j=1}^{S_\mathcal{A}} [\log p(y|\frac{1}{S_o} \sum_{i=1}^{S_o} g_k(h_w(x_a^{ji})))] - \text{KL}[q(\boldsymbol{u})||p(\boldsymbol{u})], \tag{17}$$

$$\text{with } x_a^{ij} \sim p(x_a|x, \phi). \tag{18}$$

The benefits of our new sample based bound are threefold: It broadens model specification, avoids hand-picking and fixing the artificial Gaussian likelihood variance, and doubles training speed.

	Model	Likelihd.	Test acc.
M1	NN	Softmax	0.9433
M2	Non-inv. Shallow GP	Gaussian	0.8357
M3	Non-inv. Shallow. GP	Softmax	0.7918
M4	Inv. Shallow GP	Gaussian	0.9516
M5	Inv. Shallow. GP	Softmax	0.9316
M6	Non-inv. Deep Kernel GP	Gaussian	0.9387
M7	Non-inv. Deep Kernel GP	Softmax	0.9351
M8	Inv. Deep Kernel GP	Gaussian	**0.9896**
M9	Inv. Deep Kernel GP	Softmax	0.9867

Table 1: Test accuracies on rotated MNIST. Invariant models outperform non-invariant counterpart. So do deep kernels contra shallow ones. The invariant deep kernel GPs perform best, outperforming state-of-the-art of 0.989 for learned invariance (Benton et al., 2020).

6.1 Rotated MNIST

The rotated MNIST dataset[6] was generated from the original MNIST dataset by randomly rotating the images of hand-written digits between 0 and 2π radians. It consists of a training set of 12.000 images along with 50.000 images for testing. We pretrain the neural network from Sec. 5.1 on rotated MNIST (Table 1, M1) and proceed as outlined in Sec. 5. As discussed in Sec. 5, we do not have guarantees that the ELBO acts as a good model selector for the neural network hyperparameters. We thus use a validation set (3000 of the 12000 training points) to find hyperparameters for the NN updates. Once a good training setting is found we re-train on the entire training set (see Appendix for settings). Fig. 8 shows the learned invariances (we use the full ϕ parameterisation but only plot rotation and x-translation for brevity). Both Gaussian and Softmax models learn to be rotation-invariant close the full 2π rotations present in the data. Table 1 contains test accuracies. Deep kernel GPs outperform their shallow counterparts by large margins (differences in test accuracy of ≥ 10 percent points). The same is true for invariant compared to non-invariant models (≥ 3 percent points). While both likelihoods achieve similar test accuracies, we observe a $2.3\times$ speedup per iteration in training for the sample-based Softmax over the Gaussian model. (Gaussian model: 2.64 seconds per iteration, Gaussian + sample bound: 1.32 sec./iter., Softmax + sample bound: 1.13 sec./iter. All runs are executed on 12 GB Nvidia Titan X/Xp GPUs.)

6.2 PatchCamelyon

The PatchCamelyon (PCam, CC0 License, Veeling et al. (2018)) dataset consists of histopathology scans

[6] https://sites.google.com/a/lisa.iro.umontreal.ca/public_static_twiki/variations-on-the-mnist-digits

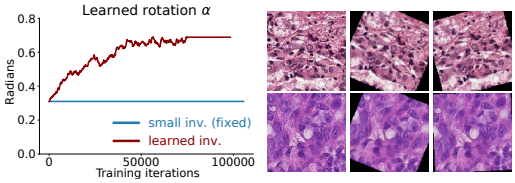


Figure 9: *Left:* Learned rotation on PCam. *Right:* PCam orbit samples. Augmented images look smoother due to interpolation, thus we preprocess the dataset with small rotations when learning invariances.

Model	Test acc.
NN	0.7905
Deep Kernel GP + no inv.	0.8018
NN + small inv.	0.7420
Deep Kernel GP + small inv.	0.8115
Deep Kernel GP + learned inv.	**0.8171**

Table 2: PCam results. InvDKGP performs best.

of lymph nodes measuring $96 \times 96 \times 3$ pixels. Labels indicate whether the centre patch contains tumor pixels. Veeling et al. (2018) improve test performance from 0.876 to 0.898 by using a NN which is invariant to (hard-coded) 90° rotations of the input. Such discrete, non-differentiable augmentations are not compatible with our backprop-based method, so we instead use continuously sampled rotations (a special case of the transformations described in Sec. 3.3 with $\phi = \alpha$ and $\alpha_{min} = \text{-}\alpha_{max}$). This, contrary to Veeling et al. (2018)'s approach, introduces the need for padding and interpolation (see Fig. 9, left), effectively changing the data distribution. We thus apply small rotations as a preprocessing step ($\pi/10$ radians, 'small inv.' in Table 2). This lowers performance for a NN alone, i.e. when pre-training. The invariant models counterbalance this performance drop, and the learned invariances produce the best results in our experiments; however, they remain subpar to Veeling et al. (2018). This is due to the limitation to differentiable transformations, as well as our simpler NN (see Appendix). We highlight that our task is fundamentally different: instead of hard-coding invariances we *learn* those during optimisation.

7 EXPLORING LIMITATIONS

Rotated MNIST and PCAM are relatively simple datasets that can be modelled using small NNs. To investigate whether our approach can be used on more complex datasets, we turn to CIFAR-10 (Krizhevsky, 2009), which is usually trained with larger models and data augmentation. Unfortunately, we found that we were unable to learn invariances for CIFAR-10.

To understand why, we designed a simple experiment. We first pretrain ResNet-18-based (He et al., 2016) networks with different levels ν of invariance transformations (see the Appendix for a definition of ν). We then train sparse GP regression (SGPR; Titsias, 2009) models on an augmented training set created by propagating ten points sampled from the augmentation distribution through these neural networks. The samples are generated at different levels of invariance ν, not necessarily matching the levels of the pretrained NNs. We plot the results in Fig. 10: when the network is trained at a small invariance level ϵ, the performance of the SGPR model is highest at an invariance level of 0.01, and rapidly drops off for larger invariances (note the logarithmic x scale). We see a similar result for the network trained at a level of 0.1. Finally, when the network is trained at the same level that the orbit points for the SGPR model are sampled at ('adapted'), we see that added invariance helps the accuracy, with no steep drop off in accuracy for larger invariances. Therefore, adding invariance does help, but only when the network has already been adapted to that invariance. Currently, in our method this coadaptation is prevented by the current need for coordinate ascent training (Sec. 5).

This experiment indicates that for datasets requiring larger neural networks, we are in a difficult position. We need to adapt the feature extractor jointly with the invariances. However, this approach leads to pathologies as the neural network parameters are not protected from overfitting (Ober et al. (2021), see Sec. 5), which we previously mitigated with coordinate ascent. Therefore, relying on the marginal likelihood to learn invariances with a large feature extractor can easily lead to unwanted behavior – this behavior prevents us from learning these invariances as easily as the marginal likelihood promises. We believe that ongoing research in Bayesian deep learning will alleviate this problem. Bayesian neural networks with methods for marginalising over lower layers too, thus protecting them against overfitting, will render our approach more easily applicable. Such advances will allow us to learn invariances more easily and on more complex tasks than we did for the MNIST and PCAM datasets.

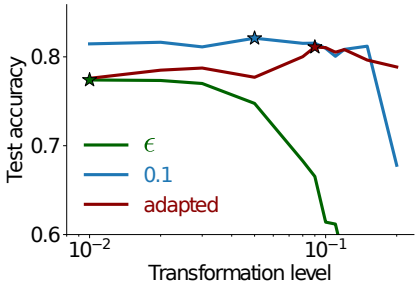


Figure 10: Test accuracies on CIFAR-10 for different transformation levels ν with pretraining at negligible "ϵ", 0.1, and adapted levels. The maxima for each curve are marked with a star, and occur at test accuracies of 77.4% for "ϵ", 82.1% for 0.1, and 81.1% for adapted levels.

8 CONCLUSION

Neural networks depend on good inductive biases in order to generalise well. Practitioners usually – successfully – handcraft inductive biases, but the idea of learning them from data is appealing. Might we automate the modelling pipeline, moving from handcrafted models to data driven models; much like we replaced hand-crafted features with learned features in deep neural networks? This work proposes one step in this direction. Inspired by Bayesian model selection we employ the marginal likelihood for learning inductive biases. We avoid the intractability of the marginal likelihood for neural networks by using Deep Kernel Learning. This enables us to leverage previous work on invariance learning in GPs for learning data augmentation in neural networks. We learn useful invariances and improve performance, but encounter challenges when optimising our models. We introduce a new sampling-based bound to the ELBO allowing for inference for the Softmax likelihood, the natural choice for classification tasks, hereby alleviating some of the optimisation difficulties. Others we identify as fundamental limitations of the Bayesian last layer approach.

Societal Impact: This work is situated within basic research in probabilistic ML and, as such, bears all the risks of automation itself: harmful redistribution of wealth to those with access to compute resources and data, loss of jobs, and the environmental impact of such technologies. In fact, our model is more computationally heavy than a standard neural network with hand-tuned data augmentation. However, in the long term, automatic model selection has the potential to *reduce* the need for hyperparameter tuning, which usually dramatically exceeds the resources needed for training the final model.

Acknowledgements

MJ is supported by a research grant from the Carlsberg Foundation (CF20-0370). SWO acknowledges the support of the Gates Cambridge Trust for his doctoral studies.

References

G. Benton, M. Finzi, P. Izmailov, and A. G. Wilson. Learning invariances in neural networks. In *Advances in Neural Information Processing Systems*, 2020.

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

J. Bradshaw, A. G. d. G. Matthews, and Z. Ghahramani. Adversarial examples, uncertainty, and transfer testing robustness in Gaussian process hybrid deep networks. *arXiv preprint arXiv:1707.02476*, 2017.

T. D. Bui, J. Yan, and R. E. Turner. A unifying framework for Gaussian process pseudo-point approximations using power expectation propagation. *Journal of Machine Learning Research (JMLR)*, 18 (104):1–72, 2017.

Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. 2016.

D. R. Burt, C. E. Rasmussen, and M. van der Wilk. Convergence of sparse variational inference in Gaussian processes regression. *Journal of Machine Learning Research (JMLR)*, 21(131):1–63, 2020.

R. Calandra, J. Peters, C. E. Rasmussen, and M. P. Deisenroth. Manifold Gaussian processes for regression. In *International Joint Conference on Neural Networks (IJCNN)*, 2016.

E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020.

A. Damianou and N. D. Lawrence. Deep Gaussian processes. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2013.

T. Dao, A. Gu, A. Ratner, V. Smith, C. De Sa, and C. Ré. A kernel theory of modern data augmentation. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

V. Dutordoir, M. van der Wilk, A. Artemev, and J. Hensman. Bayesian image classification with deep convolutional Gaussian processes. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

V. Dutordoir, J. Hensman, M. van der Wilk, C. H. Ek, Z. Ghahramani, and N. Durrande. Deep neural networks as point estimates for deep Gaussian processes. In *Advances in Neural Information Processing Systems*, 2021.

D. Ginsbourger, X. Bay, O. Roustant, and L. Carraro. Argumentwise invariant kernels for the approximation of invariant functions. In *Annales de la Faculté des sciences de Toulouse: Mathématiques*, volume 21, pages 501–527, 2012.

D. Ginsbourger, N. Durrande, and O. Roustant. Kernels and designs for modelling invariant functions: From group invariance to additivity. In *mODa 10–Advances in Model-Oriented Design and Analysis*, pages 107–115. Springer, 2013.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

J. Hensman, A. G. d. G. Matthews, and Z. Ghahramani. Scalable variational Gaussian process classification. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

G. E. Hinton and R. Salakhutdinov. Using deep belief nets to learn covariance kernels for Gaussian processes. *Advances in Neural Information Processing Systems*, 2007.

D. Ho, E. Liang, X. Chen, I. Stoica, and P. Abbeel. Population based augmentation: Efficient learning of augmentation policy schedules. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

A. Immer, M. Bauer, V. Fortuin, G. Rätsch, and M. E. Khan. Scalable marginal likelihood estimation for model selection in deep learning. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR)*, 2015.

I. R. Kondor. *Group theoretical methods in machine learning*. Columbia University, 2008.

A. Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Y. LeCun, C. Cortes, and C. J. Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*.

J. Lorraine, P. Vicol, and D. Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *Proceedings of the 23rd Conference on*

International Conference on Artificial Intelligence and Statistics (AISTATS), 2020.

D. J. C. MacKay. Model comparison and Occam's razor. *Information Theory, Inference and Learning Algorithms*, pages 343–355, 2003.

A. G. d. G. Matthews, J. Hensman, R. E. Turner, and Z. Ghahramani. On sparse variational methods and the Kullback-Leibler divergence between stochastic processes. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

A. G. d. G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani, and J. Hensman. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research (JMLR)*, 18(40):1–6, 2017.

M. Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, 1961.

S. Nabarro, S. Ganev, A. Garriga-Alonso, V. Fortuin, M. van der Wilk, and L. Aitchison. Data augmentation in bayesian neural networks and the cold posterior effect, 2021.

S. W. Ober and L. Aitchison. Global inducing point variational posteriors for Bayesian neural networks and deep Gaussian processes. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2020.

S. W. Ober, C. E. Rasmussen, and M. van der Wilk. The promises and pitfalls of deep kernel learning. In *Proceedings of the 37th Conference on Uncertainty in Artifical Intelligence (UAI)*, 2021.

C. E. Rasmussen and Z. Ghahramani. Occam's razor. *Advances in Neural Information Processing Systems*, 2001.

P. Schwöbel, F. Warburg, M. Jørgensen, K. H. Madsen, and S. Hauberg. Probabilistic spatial transformers for Bayesian data augmentation. *arXiv preprint arXiv:2004.03637*, 2020.

M. Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.

R. E. Turner and M. Sahani. Two problems with variational expectation maximisation for time-series models. In D. Barber, T. Cemgil, and S. Chiappa, editors, *Bayesian Time Series Models*, chapter 5, pages 109–130. Cambridge University Press, 2011.

J. van Amersfoort, L. Smith, A. Jesson, O. Key, and Y. Gal. On feature collapse and deep kernel learning for single forward pass uncertainty. *arXiv preprint arXiv:2102.11409*, 2021.

M. van der Wilk, M. Bauer, S. T. John, and J. Hensman. Learning invariances using the marginal likelihood. In *Advances in Neural Information Processing Systems*, 2018.

M. van der Wilk, V. Dutordoir, S. T. John, A. Artemev, V. Adam, and J. Hensman. A framework for interdomain and multioutput Gaussian processes. *arXiv preprint arXiv:2003.01115*, 2020.

B. S. Veeling, J. Linmans, J. Winkens, T. Cohen, and M. Welling. Rotation equivariant CNNs for digital pathology. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018.

C. K. I. Williams and C. E. Rasmussen. *Gaussian processes for machine learning*. MIT Press Cambridge, MA, 2006.

A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016a.

A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Stochastic variational deep kernel learning. *Advances in Neural Information Processing Systems*, 2016b.

J. T. Wilson, V. Borovitskiy, A. Terenin, P. Mostowsky, and M. P. Deisenroth. Efficiently sampling functions from Gaussian process posteriors. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

A. Zhou, T. Knowles, and C. Finn. Meta-learning symmetries by reparameterization. In *9th International Conference on Learning Representations (ICLR)*, 2021.

Supplementary Material:
Last Layer Marginal Likelihood for Invariance Learning

A IS THE MARGINAL LIKELIHOOD NECESSARY?

Sec. 1 motivated the marginal likelihood for invariance learning. Given that this loss function is notoriously difficult to evaluate, we verify experimentally that using it is indeed *necessary*, i.e. that the standard maximum likelihood loss is insufficient. Fig. 11 shows invariances learned on rotated MNIST (rotMNIST, see Sec. 6.1 for a description of the dataset) by using a neural network with maximum likelihood loss for two initialisations (blue, green). They collapse as suggested by the theory. The marginal likelihood solution (red) instead identifies appropriate invariances.

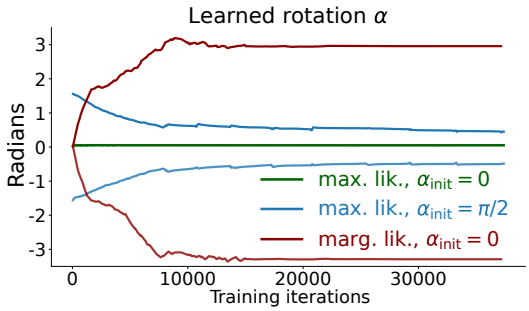


Figure 11: Max. likelihood (green, blue, *collapsing*) and marg. likelihood (red, *useful*) invariances.

B EXPERIMENTAL DETAILS

Exploiting the ideas from Sec. 5, we start by training convolutional neural networks (CNNs, see below for architecture details). After pre-training the CNN, we replace the last fully connected layer with a GP and continue training. In the non-invariant case we train all parameters jointly from here. When learning invariances, we iterate between updating the GP variational- and hyperparameters, and the neural network weights.

B.1 MNIST variations

We here summarise the training setups for the experiments on MNIST variations, i.e. MNIST subsets (Sec. 5.1) and rotated MNIST (Sec. 6.1). We start by outlining the shared neural network architecture and will then list the hyperparameter settings for MNIST and rotMNIST, respectively.

The CNN architecture used in the (rot)MNIST experiments is depicted in Table 3. For rotated MNIST, we train the model for 200 epochs with the Adam optimiser (default parameters). For the MNIST subsets, we train for 60k iterations which corresponds to 200 epochs for the full dataset and respectively more epochs for smaller subsets. The remaining parameters are the same in all experiments: batch size 200, learning rate 0.001, no weight decay, other regularisation or data augmentation. In the pre-training phase we minimise negative log-likelihood, for updates during coordinate ascent we use the ELBO as a loss-function.

Hyperparameter initialisation for the MNIST subset experiments are lengthscale 10, likelihood variance 0.05, kernel variance 1 (fixed likelihood and kernel variance for the invariant model, see Sec. 5.1), posterior variance 0.01. We use 1200 inducing points which we initialise by first passing the images through the neural

Layer	Specifications
Convolution	filters=20, kernel size=(5, 5), padding=same, activation=ReLU
Max pooling	pool size=(2, 2), stride=2
Convolution	filters=50, kernel size=(5, 5), padding=same, activation=ReLU
Max pooling	pool size=(2, 2), stride=2
Fully connected	neurons=500, activation=ReLU
Fully connected	neurons=50, activation=ReLU
Fully connected	neurons=10, activation=Softmax

Table 3: Neural network architecture for MNIST variations. After pre-training, the last fully connected layer (below dashed line) is replaced with a GP layer for the deep kernel models.

network, then using the 'greedy variance' method (Burt et al., 2020) on the extracted features. For the smallest dataset MNIST312 we use 312 inducing points only. The batch size is 200 and we choose learning rate 0.001 for the Adam optimiser. For the invariant models, the orbit size is 120 and affine parameters are initialised at $\phi_{min} = \phi_{max} = 0.02$, i.e. we initialise with a small invariance. Without this initialisation we encountered occasional numerical instabilities (Cholesky errors) on the small dataset runs. During coordinate ascent (InvDKGP models) we toggle between training GP and CNN after 25k steps.

Hyperparameter initialisation for the rotMNIST experiment as follows: For all models, we initialise kernel variance 1 (fixed at 1 for $M9$, see Sec. 5.1) and posterior variance 0.01. We use 1200 inducing points. For the invariant models, the orbit size is 120 and affine parameters are initialised at $\phi_{min} = \phi_{max} = 0$, i.e. invariances are learned from scratch. When using coordinate ascent (InvDKGP models, M8 & M9) we toggle between training GP and CNN after 30k steps. We train different models for a different number of iterations, all until the ELBO has roughly converged. Batch size 200 is used for all models. The remaining initialisations differ between models and are summarised in Table 4.

	Model	Lengthsc.	Lik. var.	LR (decay)
M2	Non-inv. Shallow GP + Gaussian	10	0.02	0.001
M3	Non-inv. Shallow. GP + Softmax	10	-	0.001
M4	Inv. Shallow GP + Gaussian	10	0.05	0.001
M5	Inv. Shallow. GP + Softmax	10	-	0.001
M6	Non-inv. Deep Kernel GP + Gaussian	10	0.05	0.001
M7	Non-inv. Deep Kernel GP + Softmax	20	-	0.001
M8	Inv. Deep Kernel GP + Gaussian	50	0.05 (F)	0.003 (steps / cyclic)
M9	Inv. Deep Kernel GP + Softmax	9	-	0.003 / 0.0003 (s / c)

Table 4: Training settings for rotMNIST models: Kernel lengthscale and likelihood variance initialisations ('F' indicates a fixed likelihood variance, see Sec. 5.1). The learning rate column ('LR') also indicates whether the learning rate was decayed in the GP/CNN update phases of coordinate ascent. For the 'steps'(s) decay, we divide by 10 after 50% and again 75% of iterations, for the 'cyclic'(c) decay, learning rates are: [LR/100, LR/10, LR, LR/10, LR/100]. These training hyperparameters are determined using a validation set (see Sec. 6.1).

B.2 PCam

The CNN architecture is a VGG-like convolutional neural network[7] described in Table 5. The model is trained for 5 epochs using the Adam optimiser with batch size 64. We use learning rate 0.001 which we divide by 10 after 50% and again 75% of training iterations. In the fully connected block we use dropout with 50% probability when pre-training. Dropout is disabled when training the deep kernel models.

Hyperparameters for the deep kernel GP experiments on PCam are: lengthscale 10 (1 for the learned invariance model), kernel variance 1, posterior variance 0.01. We use 750 inducing points which we initialise as in the previous experiments. The batch size is 32. For PCAm we use coordinate ascent for all models since

[7] We closely follow `https://geertlitjens.nl/post/getting-started-with-camelyon/`.

Layer	Specifications
Convolution	filters=16, kernel size=(3, 3), padding=valid, activation=ReLU
Convolution	filters=16, kernel size=(3, 3), padding=valid, activation=ReLU
Max Pooling	pool size=(2, 2), strides=2
Convolution	filters=32, kernel size=(3, 3), padding=valid, activation=ReLU
Convolution	filters=32, kernel size=(3, 3), padding=valid, activation=ReLU
Max Pooling	pool size=(2, 2), stride=2
Convolution	filters=64, kernel size=(3, 3), padding=valid, activation=ReLU
Convolution	filters=64, kernel size=(3, 3), padding=valid, activation=ReLU
Max Pooling	pool size=(2, 2), stride=2
Fully Connected	neurons=256, activation=ReLU
Dropout	probability=0.5
Fully Connected	neurons=50, activation=None
Dropout	probability=0.5
Fully connected	neurons=2, activation=Softmax

Table 5: Neural network architecture for PCAM. After pre-training, the last fully connected layer (below dashed line) is replaced with a GP layer for the deep kernel models and dropout is disabled.

this improves training stability. Learning rates are 0.001 for the GP update steps and 0.0001 for the CNN updates, no LR decay. We toggle between the two coordinate ascent phases after 50k and 75k iterations in the non-invariant and invariant case, respectively. For the invariant models the orbit size is 20 and we initialise the rotation invariance with $\phi_{min/max} = \alpha_{min/max} = \pm\pi/10$.

B.3 CIFAR-10

Throughout our experiments, we train on a subset of 45,000 points from the full CIFAR-10 (Krizhevsky, 2009) training set and report results on the remaining 5,000 points, as a validation set.

The model we use is a sparse GP regression (SGPR; Titsias (2009)) model with a sum kernel corresponding to a Monte Carlo estimate of the kernel of Eq. 5, using an automatic relevance determination (ARD) squared exponential kernel as a base kernel. We achieve this by sampling 10 points from the full orbit for each data point, and propagating the points through the pretrained feature extractor. For the feature extractor, we choose a ReLU ResNet-18 architecture (He et al., 2016) with an output dimension of 50, using the post-ReLU features. Therefore, for our training set, we end up with a set of $45,000 \times 10 \times 50$ datapoints, where we sum over the 10 orbit samples.

Hyperparameters were chosen as follows. We pretrain the ResNet-18 by adding an additional fully-connected layer with softmax activations. We train for 160 epochs with a batch size of 100 and the Adam optimizer (Kingma and Ba, 2015), starting with a learning rate of 0.001, which we step down by a factor of 10 at epochs 80 and 120. We train the network without weight decay. The SGPR model was subsequently trained for a maximum of 1000 steps, using the Scipy optimizer provided in GPflow (Matthews et al., 2017). During training, we initially set the jitter to 1e-6, which we increased by a factor of ten if the Cholesky decomposition failed. For the SGPR model, we use 1000 inducing points, initialised as above. We found empirically that the likelihood variance did not have a significant impact on the results; we therefore fixed it to 0.01. Recalling that $\phi = (\alpha, s^x, s^y, p^x, p^y, t^x, t^y)$, we parameterize the transformation by considering the "transformation level" ν such that

$$\phi_{\max} = (0, 1, 1, 0, 0, 0, 0) + \nu \times (\pi, 1, 1, 1, 1, 1, 1), \tag{19}$$

$$\phi_{\min} = (0, 1, 1, 0, 0, 0, 0) - \nu \times (\pi, 1, 1, 1, 1, 1, 1). \tag{20}$$

For the "ϵ" setting of the transformation level, we assign $\nu = 0.01$. We chose a non-zero value to ensure that any reduction in performance would be due to a different value of ν, and not because of the lack of presence of the image interpolator in the pretraining (see Sec. 6.2).

Paper B

Probabilistic Spatial Transformer Networks

Probabilistic Spatial Transformer Networks

Pola Schwöbel[1] **Frederik Warburg**[1] **Martin Jørgensen**[2] **Kristoffer H. Madsen**[1, 3] **Søren Hauberg**[1]

[1] Section for Cognitive Systems, DTU Compute, Technical University of Denmark, Copenhagen, Denmark
[2]Machine Learning Research Group, Department of Engineering Science, University of Oxford, Oxford, UK
[3]Danish Research Centre for Magnetic Resonance, Centre for Functional and Diagnostic Imaging and Research, Copenhagen University Hospital Hvidovre, Hvidovre, Denmark

Abstract

Spatial Transformer Networks (STNs) estimate image transformations that can improve downstream tasks by 'zooming in' on relevant regions in an image. However, STNs are hard to train and sensitive to mis-predictions of transformations. To circumvent these limitations, we propose a probabilistic extension that estimates a stochastic transformation rather than a deterministic one. Marginalizing transformations allows us to consider each image at multiple poses, which makes the localization task easier and the training more robust. As an additional benefit, the stochastic transformations act as a localized, learned data augmentation which improves the downstream tasks. We show across standard imaging benchmarks and on a challenging real world dataset that these two properties lead to improved classification performance, robustness and model calibration. We further demonstrate that the approach generalizes to non-visual domains by improving model performance on time-series data.

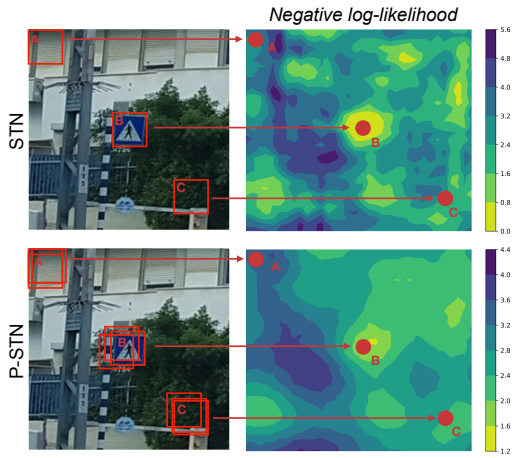


Figure 1: The Probabilistic Spatial Transformer Network (P-STN) marginalizes over a distribution of possible input transformations. By 'looking in multiple places' we hope to stabilize the brittle nature of the regular spatial transformer: The P-STN loss landscape is significantly more smooth and with fewer local minima compared to the STN.

1 INTRODUCTION

The *Spatial Transformer Network (STN)* [Jaderberg et al., 2015] predicts a *transformation* on input data in order to simplify a downstream task. For example, a neural network might benefit from e.g. 'zooming in' on relevant parts of an image, remove unwarranted image rotations, or time normalize sequence data before making predictions. In principle, this can improve robustness, interpretability and efficiency of the model. In practice, the situation is, however, not as ideal. Both at training and test time, the STN is sensitive to small mis-predictions of transformations. For example, if the STN zooms in on the wrong part of an image, then the signal is lost for the downstream task, e.g. see crop A and C in Fig. 1. The empirical impact is that STNs are difficult to train and often do not live up to their promise.

From a probabilistic perspective, this sensitivity has an obvious solution: we should estimate the posterior over the applied transformation and marginalise accordingly. This amounts to 'trying many different transformations', and should improve robustness. It is exactly this approach we investigate.

STNs consist of two parts. A localization network performs the transformation task, i.e. it estimates the transformation parameters θ for a given image I and applies the corresponding transformation $T_\theta(I)$. A standard neural network performs the downstream task on the transformed image, i.e. computing $p(y|T_\theta(I))$. Since we are concerned with

Accepted for the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022).

classification tasks we will refer to the latter as the classifier, but note that the approach generalizes to other tasks.

In our probabilistic STN (P-STN), we estimate a distribution over transformations that we marginalize: $p(y|I) = \int p(y|T_\theta(I))\mathrm{d}\theta$. We approximate this intractable integral via Monte Carlo, i.e. we sample transformations. Those transformation samples produce different transformed versions of the input image, $\{T_\theta^s(I)\}_{s=1\ldots S}$. The classifier makes predictions on all samples, and we aggregate the predictions. Figure 2 shows the model architecture.

We hypothesize that marginalizing image transformation has benefits for both parts of the model. For the *localization* network, our model gets to 'try many different transformations' through random sampling. This should improve the localization. Secondly, the classifier now gets presented with different transformed versions of the input image through Monte Carlo samples $\{T_\theta^s(I)\}_{s=1\ldots S}$. Interestingly, this corresponds to a type of data augmentation, which should improve classification.

We verify the hypothesis by making the following contributions:

1. We develop the Probabilistic Spatial Transformer; a hierarchical Bayesian model over image transformations.
2. Perform variational inference to fit the transformation model as well as downstream model end-to-end, using only label information.
3. Experimentally demonstrate that our model achieves better localization, increased classification accuracy (resulting from learned per-image data augmentation) and improved calibration.

2 RELATED WORK

Spatial transformer networks apply a spatial transformation to the input data as part of an end-to-end trained model [Jaderberg et al., 2015]. The transformation parameters are estimated from each input separately through a neural network. Most commonly, STNs implement simple affine transformations, such that the network can learn to zoom in on relevant parts of an image before solving the task at hand. STNs have shown themselves to be useful for both generative and discriminative tasks, and have seen applications to different data modalities [Jaderberg et al., 2015, Detlefsen and Hauberg, 2019, Detlefsen et al., 2018, Shapira Weber et al., 2019, Sønderby et al., 2015, Lin and Lucey, 2016, Kanazawa et al., 2016]. We propose a probabilistic extension of this idea, replacing the usual likelihood maximization with marginalization over transformations.

Bayesian deep learning aims to solve probabilistic computations in deep neural networks. Priors are put on weights and marginalized at training and test time, often yielding useful uncertainties in the posterior predictive. The required computations are in general intractable, and approaches differ mainly in the type of approximation to the weight posterior. Gal and Ghahramani [2016] propose to view dropout as a Bernoulli approximation to the weight posterior (i.e. randomly switching each weight on or off). The Laplace approximation [MacKay, 1992, Daxberger et al., 2021] places a Gaussian posterior over a trained neural network's weights. Another generally successful way to obtaining predictive uncertainties is to simply train an ensemble of models. Originally proposed as an alternative to Bayesian DL [Lakshminarayanan et al., 2017], the approach can be interpreted in the Bayesian framework by interpreting the weights of the trained ensemble members as samples from a weight posterior [Gustafsson et al., 2020]. Similar to our method, Blundell et al. [2015] choose a variational approach with a simple Gaussian mean field posterior over weights. Our approach differs from standard Bayesian DL in that we are not reasoning about distributions over neural network weights $p(w)$, but instead a subnetwork's (i.e. the localizer's) *outputs* $p(\theta)$. Drawing from the posterior over image transformations, we effectively recover data augmentation.

Data augmentation (DA) is an useful way to increase the amount of available data [LeCun et al., 1995, Krizhevsky et al., 2012]. DA requires prior knowledge about the structure of the data: the target y is assumed to be invariant to certain transformations of the observation I. Invariance assumptions are usually straight forward for natural images. Thus, DA is common for image data, where the transformation family is often chosen to be rotations, scalings, and similar [Goodfellow et al., 2009, Baird, 1992, Simard et al., 2003, Krizhevsky et al., 2012, Loosli et al., 2007]. The general trend is that, beyond 'intuitive' data such as images, gathering an invariance prior is difficult, and DA is often hard to realize through manual tuning.

Learned data augmentation provides a more principled approach to artificially extending datasets. Hauberg et al. [2016] estimate an augmentation scheme from the training data via pre-aligning images in an unsupervised manner. The approach allows for significantly more complex transformations than the usual affine family, but the unsupervised nature and the implied two step training process render the approach suboptimal. Similarly, Cubuk et al. [2019, 2020] use reinforcement learning and grid search to learn data augmentation schemes, but rely on on validation data rather than an end-to-end formulation.

Learning data augmentation end-to-end requires a loss function suitable for model selection as we are effectively trying to learn an inductive bias. Based on this realization, Van der Wilk et al. [2018] learn DA end-to-end in Gaussian processes (GPs) via the marginal likelihood, a suitable loss for model selection and thus invariance learning [MacKay, 2003]. The marginal likelihood is hard to compute for NNs, so Schwöbel et al. [2022] extend this idea to NNs by consid-

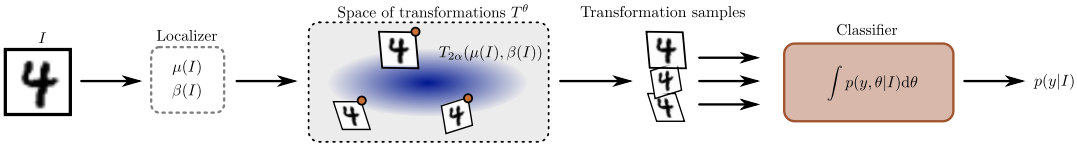


Figure 2: The P-STN pipeline. From the observed image I, a distribution of transformations is estimated. Samples from this distribution are applied to the observed image to produce augmented samples, which are fed to a classifer that averages across samples. In the deterministic STN case, the localiser only computes one transformation $\theta(I)$, which can be thought of as the maximum likelihood solution. Instead of the multiple transformation samples, we obtain a single $T_\theta(I)$ in this case.

ering a deep kernel model, i.e. a neural network with a GP in the last layer. Benton et al. [2020] instead use the standard, maximum likelihood loss and explicitly regularize towards non-zero augmentations. Our model differs from existing data augmentation approaches — learned and non-learned — in that we estimate local, i.e. *per-image* transformations instead of a global augmentation scheme.

3 BACKGROUND

The STN localiser module estimates a transformation $\theta(x)$ that transforms a coordinate grid and interpolates an image accordingly. The classifier module takes the transformed image and computes $p(y|T_\theta(x))$. Both the localiser and classifier are neural networks. The STN can be trained end-to-end with only label information as long as the image transformations are parameterized in a differentiable manner.

Affine transformations are a simple class of transformations that can be differentiably parameterized. We limit ourselves to the subset of affine transformations containing rotation, isotropic scaling and translation in x and y. In two dimensions (and the corresponding three-dimensional homogeneous coordinates), we thus learn $\theta = (r, s, t_x, t_y)$ which parameterizes the affine matrix

$$A_\theta = \begin{bmatrix} s \cdot \cos r & -s \cdot \sin r & t_x \\ s \cdot \sin r & s \cdot \cos r & t_y \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{3\times 3},\ s > 0. \quad (1)$$

Since $\det(A_\theta) = s^2$, the constraint $s > 0$ ensures invertibility and can be implemented as seen in Detlefsen et al. [2018]. In practice, the STN estimates well-behaved, non-collapsing transformations without implementing the constraint explicitly. $T_\theta(I)$ is applied by transforming a grid of the target image size by A_θ and interpolating the source image at the resulting coordinates (see Jaderberg et al. [2015] for details).

Diffeomorphic transformations (i.e. transformations that are differentiable, invertible and possess a differentiable inverse) are more general than affine transformations, and are not limited to the spatial domain. Freifeld et al. [2017] construct diffeomorphisms from continuous piecewise-affine velocity fields as follows. The transformation domain Ω is divided into subsets and an affine matrix is defined on each cell c of such a tessellation. Each affine matrix A_{θ_c} induces a vector field mapping each point $x \in c$ to a new position $v^{\theta_c} : x \mapsto A_{\theta_c} x$. These velocity fields are then integrated to form a trajectory for each image point x

$$\phi^\theta(x; 1) = x + \int_0^1 v^\theta(\phi(x; \tau))\mathrm{d}\tau.$$

Given boundary and invertibility constraints [Freifeld et al., 2017], such a collection of affine matrices $\{A_{\theta_c}\}_{c\subset\Omega}$ defines a diffeomorphic transformation $T^\theta : x \mapsto \phi^\theta(x, 1)$.

The libcpab library [Detlefsen, 2018] provides an efficient implementation for this approach, specifically optimized for use in a deep learning context where fast gradient evaluations are crucial. The author successfully employs CPAB-transformations within a Spatial Transformer Network [Detlefsen et al., 2018].

4 PROBABILISTIC SPATIAL TRANSFORMER

The P-STN is a probabilistic extension of the STN, where we replace the deterministic transformation $\theta(I)$ with a posterior over transformations $p(\theta|I)$. Figure 2 illustrates the proposed pipeline. We assume observed data of the form $\mathcal{D} = \{y_i, I_i\}_{i=1}^N$, where y is the target variable (e.g. class label), and I are observations of the covariates. For presentation purposes, we will consider the latter to be images, but the approach applies to any spatio-temporal data.

4.1 THE MODEL

Recall that STNs are trained end-to-end for the downstream task using only label information. Thus, while we observe y, θ is a latent variable. We model it to be governed by a second latent variable λ. λ is a precision parameter, effectively stopping the localization distribution (i.e. the amount of 'data augmentation' we introduce) from collapsing. The necessity for non-collapsing augmentation is discussed in Benton et al. [2020], Van der Wilk et al. [2018] and Schwöbel et al. [2022].

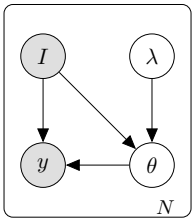


Figure 3: A graphical representation of the model structure. Grey nodes are observables and white are latents.

We wish to infer the latent variables in a Bayesian manner. This entails computing the (log-)marginal likelihood of the observed

$$\log p(I, y) = \log \iint p(I, y, \theta, \lambda)\, \mathrm{d}\theta\, \mathrm{d}\lambda. \tag{2}$$

We let the joint distribution factorise as (see Fig. 3)

$$p(y, I, \theta, \lambda) = p(y|I, \theta, \lambda)p(I, \theta, \lambda) \tag{3}$$
$$= p(y|I, \theta)p(\theta|\lambda, I)p(\lambda)p(I). \tag{4}$$

Notice $p(I)$ is unaffected by model parameters λ and θ, and in this sense can be specified without affecting the model. We define the prior over (θ, λ) semi-empirical as the prior over θ depends on observed covariates in the following way

$$p(\theta|\lambda, I) = \mathcal{N}(\theta|\mu(I), 1/\lambda), \tag{5}$$

where $\mu(I)$ is a function parametrised by a neural network, i.e. $\mu(I) := \mu_\Phi(I)$ for model parameters Φ. The prior over λ is a Gamma distribtion, i.e.

$$p(\lambda_i) = \Gamma(\alpha_0, \beta_0). \tag{6}$$

We remark here that there is one λ_i associated to each observation, and they are assumed to factorise: $p(\lambda) = \prod_{i=1}^N p(\lambda_i)$. This choice of conjugate priors for variance estimation is similar to [Stirn and Knowles, 2020, Takahashi et al., 2018, Detlefsen et al., 2019]. Finally, we assume that, conditional on I and θ, we have marginal independence in y, i.e. $p(y|I, \theta) = \prod_{i=1}^N p(y_i|I_i, \theta_i)$.

4.2 VARIATIONAL APPROXIMATION

The integral equation (2) for the marginal likelihood is intractable and, thus, the posterior $p(\lambda, \theta|I, y)$ is too. We derive a lower bound on the log marginal likelihood to utilise variational inference [Blei et al., 2017]. We choose the variational approximation q of the posterior $p(\theta, \lambda|I, y)$ as

$$q(\theta, \lambda) := p(\theta|\lambda, I)q(\lambda). \tag{7}$$

Here $p(\theta|\lambda, I)$ is given as before and $q(\lambda) := \prod_{i=1}^N \Gamma(\alpha_i, \beta(I_i))$. In our approximation, β is a neural network: hence, we use amortized inference in a similar way to the VAE model [Kingma and Welling, 2014].

We derive our lower bound using Jensen's inequality

$$\log p(y, I) = \log \iint p(y, I, \theta, \lambda)\mathrm{d}\theta\mathrm{d}\lambda \tag{8}$$
$$\geq \iint \log\left(\frac{p(y, I, \theta, \lambda)}{q(\theta, \lambda)}\right) q(\theta, \lambda)\mathrm{d}\theta\mathrm{d}\lambda \tag{9}$$
$$= \iint \log\left(\frac{p(y|I, \theta)p(\lambda)p(I)}{q(\lambda)}\right) p(\theta|\lambda, I)q(\lambda)\mathrm{d}\theta\mathrm{d}\lambda$$
$$= \underbrace{\mathbb{E}_{q(\theta,\lambda)} \log p(y|I, \theta)}_{\textbf{classification loss}} + \log p(I) - \mathrm{KL}(q(\lambda)\|p(\lambda))\,. \tag{10}$$

Thus, our evidence lower bound (ELBO) objective function (10), consists of two terms: a classification loss and a KL-term controlling the distance of the approximate posterior to the prior. During inference we can disregard $\log p(I)$ as it does not depend on parameters of interest.

4.3 INFERENCE

The choice of variational posterior implies the following for the **classification loss**

$$\mathbb{E}_{q(\theta,\lambda)} \log p(y|I, \theta) \tag{11}$$
$$= \iint \log p(y|I, \theta)q(\theta, \lambda)\mathrm{d}\theta\mathrm{d}\lambda \tag{12}$$
$$= \iint \log p(y|I, \theta)p(\theta|\lambda, I)q(\lambda)\mathrm{d}\theta\mathrm{d}\lambda \tag{13}$$
$$= \int \log p(y|I, \theta) \int \mathcal{N}(\theta|\mu(I), \lambda)\Gamma(\lambda|\alpha, \beta(I))\mathrm{d}\lambda\, \mathrm{d}\theta$$
$$= \int \log p(y|I, \theta) t_{2\alpha}(\theta|\mu(I)), \tfrac{\beta(I)}{\alpha})\mathrm{d}\theta. \tag{14}$$

Here t denotes a scaled and location-shifted Student's t-distribution with mean $\mu(I)$, scaling β, and α degrees of freedom. For clarity, the marginalized $q(\theta)$ is t-distributed. Here $p(y|I, \theta)$ is what previously was referred to as $p(y|T_\theta(I))$, i.e. the classifier conditioned the transformed I.

We approximate Eq. 14 using an unbiased estimate

$$\mathbb{E}_{q(\theta,\lambda)} \log p(y_i|I_i, \theta_i) \approx \frac{1}{S}\sum_{s=1}^S \log p(y_i|I_i, \theta_{i,s}), \tag{15}$$
$$\text{with } \theta_{i,s} \sim t_{2\alpha_i}(\cdot|\mu(I_i), \alpha_i, \beta(I_i)) \tag{16}$$

and backpropagate through neural networks $\mu(I)$ and $\beta(I)$ with the reparametrization trick. In all experiments $\alpha_i = 1$.

Combining terms, the final ELBO we maximize becomes

$$\mathcal{L}_{p,q}(I, y) \approx \sum_{i=1}^N \frac{1}{S}\sum_{s=1}^S \log p(y_i|I_i, \theta_{i,s}) - \mathrm{KL}\left(q(\lambda)||p(\lambda)\right) + \text{const}, \tag{17}$$

which is readily optimized using any gradient-based method. The KL-term is analytically tractable and differentiable between two gamma distributions.

In practice, following Higgins et al. [2016] we introduce a weight parameter w to the KL-term. This requires us to tune w but in turn makes the model robust to the choice of prior. We perform a grid-search on a validation set to find the optimal w. Alternative to this, we could have done a grid search over β_0; instead we choose $\alpha_0 = \beta_0 = 1$ for all experiments. Similar to Kingma and Welling [2014], we find it often sufficient to draw only $S = 1$ samples during training. Note that our model naturally implies marginalization, and correspondingly data augmentation, at *test-time* as well as the usual training time. At test time, we draw $S = 10$ transformation samples.

5 EXPERIMENTS & RESULTS

Our model consists of two parts, the classifier $p(y|T_\theta(I))$ and the probabilistic localiser estimating the distribution over transformations. In the following experiments, we aim to disentangle our model's benefits for localization (Sec. 5.1), classification (Sec. 5.2) and calibration (Sec. 5.3).

The probabilistic localiser estimates $q(\theta) = t_2(\theta|\mu(I), \beta(I))$, i.e. in practice we implement a mean and a variance network, $\mu(I)$ and $\beta(I)$, respectively (see Fig. 2 for the architecture). We employ a small convolutional network (`Conv2d`, `Maxpool2d`, `ReLU`, `Conv2d`, `Maxpool2d`, `ReLU`) followed by two fully connected layers for both the localiser and classifier unless stated otherwise. The P-STN localiser has two heads, one for the mean and one for the variance. The number of parameters is stated in each experimental subsection. Unless stated otherwise, we keep the amount of parameters constant, i.e. when adding a localization network we remove the extra parameters from the classifier for fair comparison.

Our model is implemented in PyTorch and experiments are run on 12 GB Nvidia Titan X GPUs. The code is available at `https://github.com/FrederikWarburg/pSTN-baselines`.

5.1 MARGINALIZING TRANSFORMATIONS IMPROVES LOCALIZATION ACCURACY

The appeal of STN models is that they are trained end-to-end, i.e. based only on labels for the downstream task, and not the transformations themselves. This same property, however, is what makes the STN hard to fit. The only signal we obtain is through the supervised downstream task (i.e. the classification labels) and thus gradient information is sparse. We will now investigate whether estimating a posterior over transformations and marginalizing, i.e. 'getting to try multiple transformations', make the task easier.

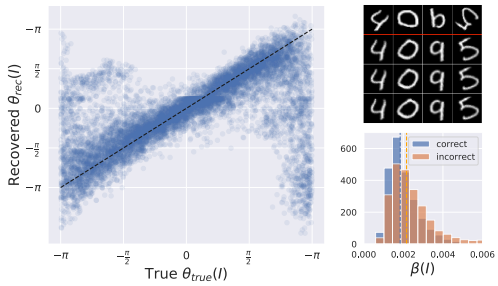


Figure 4: Rotated MNIST experiment. *Left panel:* Ground truth transformation (rotation angles in radians) against recovered transformations (mean). *Top right:* Example images from the data set and samples from the P-STN localiser. The localiser learns to pose-normalize. *Bottom right:* Outputs of the variance network. When the transformation recovery is poor (the error ε is above the median, in orange) the variances are slightly higher than when the localization works well (blue).

In order to disentangle the localization from the classification task we construct the following experiments. We first train a CNN on a pose-normalized dataset (regular MNIST and Fashion MNIST). We then generate a new dataset by randomly sampling transformations θ_{true} and applying them to the MNIST images. Saving those transformations we have ground truth available. We freeze the CNN weights and train STN and P-STN with this fixed classifier, effectively learning to recover and 'undo' the true transformations.

5.1.1 Rotated MNIST

From this data generating process we obtain a rotated version of the MNIST dataset (i.e. regular MNIST with ground truth transformations given by rotation angles, $\theta_{\text{true}}(I) = r_{\text{true}}(I)$). See Fig. 4, top right panel for example data.

Our CNN classifier (28k weights) obtains 99.4% test accuracy on MNIST and 41.2% on rotated MNIST (frozen weights, no re-training). The STN and P-STN ($S = 10$ training samples, $w = 3 \cdot 10^{-5}$, same CNN classifier as before $+72$k params in the localizer) both learn to pose-normalize, i.e. to recover these transformations to a satisfactory degree. When training the localizers only (classifier weights remain frozen as described above), the STN test acc. is 76.13%, and 82.98% for the P-STN. We compute the expected average transformation error on the $N = 10k$ rotated MNIST test images as

$$\varepsilon = \frac{1}{N} \sum_{i=1}^{N} \|\theta_{\text{true}}(I_i) - \mu(I_i)\| \mod \pi. \tag{18}$$

Figure 6: The P-STN learns to localize traffic signs in the challenging MTSD dataset. At test time, we sample 10 transformations as shown with the various bounding boxes overlaid the images. These learned variations improves the final classification.

	Acc. ↑	NLL ↓
CNN	76.0	0.49
STN	90.6	0.31
P-STN	**92.2**	**0.29**

Table 1: Accuracy (Acc.) and negative log-likelihood (NLL) for CNN, STN and P-STN.

We get $\varepsilon = 0.76$ for the STN and $\varepsilon = 0.59$ for the P-STN. The P-STN outperforms the STN, i.e. modeling uncertainty in the transformations helps in the localization task.

Uncertainty. The bottom right panel of Fig. 4 shows a histogram of $\beta(I)$, i.e. the localiser variance (or, correspondingly, the magnitude of augmentation) per image. In orange, we plot variances for images where pose-normalization is difficult (the transformation error ε is larger than the median). In blue, we plot variances for images that are correctly pose-normalized (transformation error ε smaller than the median). The poorly localized images are, on average, assigned 17% larger variances $\beta(I)$. The localiser uncertainty and thus the amount of data augmentation applied is somewhat meaningful, corresponding to the difficulty of the task.

5.1.2 Random placement FashionMNIST

We repeat a similar experiment on the slightly more challenging FashionMNIST dataset [Xiao et al., 2017] . The CNN baseline accuracy is 90.63% (same model as above with 28k parameters). We then randomly sample an x and y coordinate and place the FashionMNIST accordingly on a black background, after downscaling it by 50%. No rotation is applied, i.e. $\theta_{\text{true}} = [0, 0.5, t^x_{\text{true}}, t^y_{\text{true}}]$.

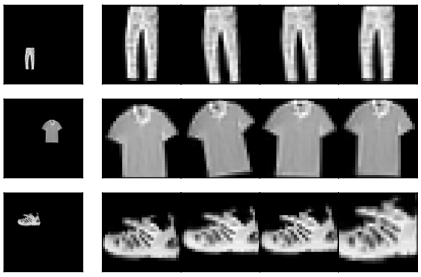


Figure 7: Random Placement Fashion MNIST. Input images (left) and transformed samples $T_{\theta_s}(I)$ as learned by the P-STN. The P-STN learns to correctly pose-normalize and zoom in to the relevant part of the image. The samples look like plausible candidates for a data augmentation scheme, this we will explore in Sec. 5.2.

Like in the previous experiment, both localizers successfully recover θ_{true}, with the P-STN ($S = 10$ training samples, $w = 3e{-}05$, same classifier as before $+193$k weights in the localizer) doing slightly better than its deterministic counterpart: test accuracies are 84.99% and 84.41%, respectively. Inspecting the transformation posterior and the resulting samples $T_{\theta_s}(I_i)$ we find that those look visually pleasing, and, as hypothesized, might be promising candidates for a data augmentation scheme. We will explore this in Sec. 5.2.

5.1.3 Mapillary street signs

Detection and classification of objects in images have many applications, e.g. for autonomous vehicles traffic signs detection is crucial. We compare a top performing classifier, a STN and our P-STN on the challenging Mapillary Traffic Sign Dataset (MTSD) [Ertler et al., 2019].

To focus this comparison, we select images that contain only one traffic sign. We obtain this subset by selecting all bounding boxes that do not intersect with other bounding boxes plus a margin of 150 px to each side. We further select the ten most common classes from this subset. This gives us a training set of 4698 images and test set of 500 images. Figure 6 shows examples images from the chosen subset.

Our classifier is a ResNet18 pre-trained on ImageNet, where we replace the last fully connected layer. We use the same ResNet for the localizers in the STN and P-STN, where we similarly replace the last layer. As before, we wish to study the behavior of the localizers. Therefore, we again start by training a classifier on the ground-truth bounding boxes. We then initialize the classifier module of the STN and P-STN with this pre-trained classifier and freeze the weights of the classifier. We train the localizers of the STN and P-STN for 60 epochs with learning rate 10^{-4} and kl weight $w = 10^{-7}$. Figure 6 shows that the P-STN learns to localize the traffic signs. At test time, we sample 10 transformations illustrated by the multiple overlaying bounding boxes.

Table 1 shows that both the STN and P-STN clearly outperform the baseline classifier when trained on the full images. Even though, the STN and P-STN have exactly the same classifier, the P-STN achieves better performance because of the ensemble of classified transformations.

	MNIST30	MNIST100	MNIST1000	MNIST3000	MNIST10000
CNN	70.12 ± 2.46	87.29 ± 0.58	95.80 ± 0.33	**97.48** ± 0.21	**97.82** ± 0.34 -
affine STN	69.26 ± 4.53	82.16 ± 2.30	92.05 ± 0.58	94.71 ± 0.22	96.96 ± 0.20
affine P-STN	**81.00** ± 3.92	**92.70** ± 0.74	**96.62** ± 0.58	**97.33** ± 0.17	**97.63** ± 0.23
optimal w	0.001	0.0003	0.0001	0.00003	0.00001

Table 2: The performance of a CNN, STN and P-STN on differently sized MNIST datasets. Bold numbers indicate that a model is significantly better than the runner up under a two sample t-test at $p = 0.05$.

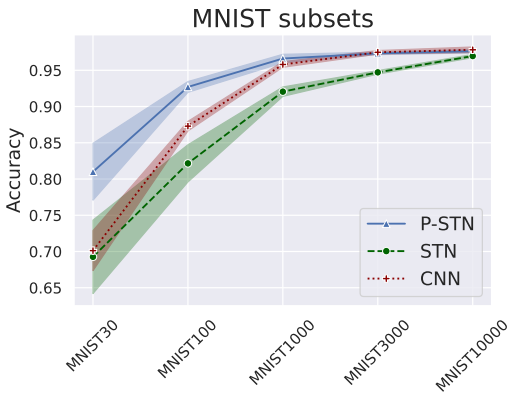


Figure 8: Performances of P-STN, STN and CNN on MNIST subsets (mean $\pm$ one STD across five folds).

5.2 MARGINALIZING TRANSFORMATIONS IMPROVES CLASSIFICATION ACCURACY

We have argued that marginalizing transformations via samples corresponds to learned, localized data augmentations (the samples $T_{\theta_s}(I)$). We will now investigate whether these augmentations are indeed helpful in the downstream task, i.e. whether they improve classification performance.

5.2.1 MNIST and subsets

We compare the performance of our P-STN against a standard convolutional neural network (CNN) and a regular STN on MNIST. The standard MNIST images are centered and pose-normalized, so the localization task is easy. Improved classifier performance can thus be viewed as an indicator for having learned a useful data augmentation scheme.

Data augmentation is particularly important when training data is scarce, so we evaluate the models on small subsets of MNIST: MNIST30 contains 30 images (i.e. 3 per class), MNIST100, MNIST1000, MNIST3000 and MNIST10000. STN and P-STN parameterize affine transformations, i.e. the learned θ is interpreted as the full affine matrix as described in Sec. 3. All models have roughly 28k parameters, architecture as described at the top of Sec. 5. We use the Adam optimizer with weight decay 0.01 and the default parameters of its PyTorch implementation. The images are color normalized. We repeat the experiment 5 times, each time with a different k-image subset of the MNIST dataset, and we report $\pm$ one standard deviation in tables and error bars. From Table 2 and Fig. 8, we see that the P-STN outperforms both the STN and CNN on the small dataset sizes. For the larger datasets the differences vanish. This supports our hypothesis: data augmentation is especially useful when data is a limited resource. This intuition is also supported by the optimal KL-weights (Table 2, bottom row) that we determine via grid search on validation data. For smaller datasets, larger w and thus more regularization towards the variance prior (away from 0) are beneficial.

The fact that the STN performs less well than the standard CNN on this data set might be explained by the fact that the images are already nearly perfectly pose-normalized, and wrong transformations can be detrimental.

5.2.2 UCR time-series dataset

For some data modalities, such as time-series, it is not trivial to craft an useful data augmentation scheme. In this experiment, we show that the P-STN can learn an useful, non-trivial data augmentation scheme that increases performance compared to a standard STN on time-series data. The UCR dataset [Dau et al., 2018] is composed of 108 smaller datasets, where each dataset contains univariate time-series. The FordA dataset, for example, contains measurements of engine noise over time and the goal is to classify whether or not the car is faulty. We select 5 of those subsets, each large enough to divide into training and validation set ($75/25\%$), which we use to find the optimal w via grid-search; those are $[0.0001, 1e-05, 0.001, 0.0, 0.0001]$. We draw $S = 10$ training samples. The test-set is pre-defined by the dataset curators. Learning rate and optimizer are the same as in Sec. 5.2.1, but we do not perform normalization. All models have approximately one million parameters. Table 3 shows that the P-STN achieves higher mean accuracy than both the STN and the CNN, indicating that we can automatically learn an useful data augmentation scheme for time-series.

We verify this qualitatively in Fig. 10 which shows an exam-

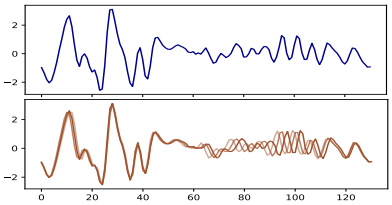


Figure 10: Examples of augmentations for a time-series from the FaceAll dataset. The top plot shows the original time-series and the bottom plot shows three augmented versions of the time-series.

	CNN	STN	P-STN
FaceAll	80.83 ± 0.62	82.28 ± 0.42	**84.31** ± 0.75
TwoPatterns	97.92 ± 0.53	99.79 ± 0.04	**99.96** ± 0.04
wafer	**99.63** ± 0.05	99.18 ± 0.17 -	98.86 ± 0.20
uWaveGestureLib.*	74.15 ± 1.27	79.77 ± 0.42 -	**81.13** ± 0.46
PhalangesOutlC.**	79.88 ± 1.32	**82.26** ± 0.98	**81.66** ± 0.59
Mean	86.48	88.65	**89.18**

Table 3: Accuracies on a subset of the UCR timeseries dataset (full dataset names are *uWaveGestureLibrary and **PhalangesOutlinesCorrect). ± 1 STD is reported after 5 repetitions. Bold numbers indicate that a model is significantly better than the runner up under a two sample t-test at $p = 0.05$.

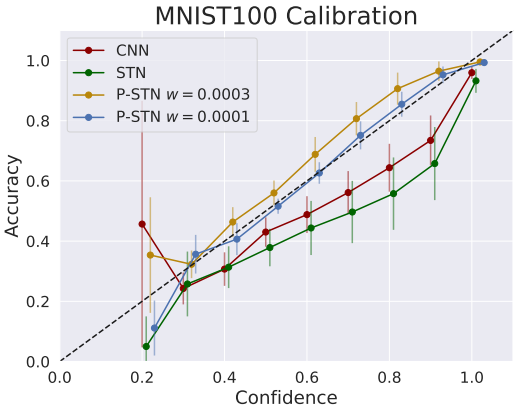


Figure 11: Calibration plots for CNN, STN and two P-STN models. One with KL-weight yielding optimal performance ($w = 0.0003$) and one with KL-weight yielding optimal calibration ($w = 0.0001$). Both P-STN models are better calibrated than CNN and STN.

ple of the learned data augmentation. We see that the model does not simply apply a global transformation, but learns to augment the time-series more in some intervals, such as in $[60; 110]$, and augment the time-series less in other intervals, such as in $[0; 50]$.

5.3 MARGINALIZING TRANSFORMATIONS IMPROVES CALIBRATION

In Sec. 5.1 we have seen that harder images on average have larger transformation uncertainties. We now investigate whether those meaningful localization uncertainties translate into meaningful uncertainties downstream, i.e. in the calibration of our classifier. At test-time, we evaluate

$$p(y|I) = \int p(y|I, \theta) q(\theta) \mathrm{d}\theta \approx \frac{1}{S} \sum_{s=1}^{S} p(y|T_{\theta_s}(I)), \quad (19)$$

i.e. we will investigate how well the uncertainty in this distribution matches the quality of predictions. Fig. 11 shows a calibration plot for the MNIST100 subset classification task from Sec. 5.2.1 for the CNN, STN and P-STN for two different w-parameters; $w = 0.0003$ yields the best performance (reported in Table 2) and $w = 0.0001$ yields the best calibration. The expected calibration errors [Guo et al., 2017, Küppers et al., 2020, 2021] are CNN: 0.0743 ± 0.0094, STN: 0.1160 ± 0.0205, P-STN, $w = 0.0003$ (optimal performance model): 0.0567 ± 0.0065, P-STN, $w = 0.0001$ (optimal calibration model): **0.0271** ± 0.0088. We report the mean over 5 folds, $\pm$ one STD. The P-STN significantly improves calibration in the downstream classification task.

5.4 A TYPICAL FAILURE MODE IN STNS

STN are trained end-to-end, and with only label information available. Thus, one aims to learn the transformation which is optimal for solving the downstream task. Depending on the complexity of the downstream task and the classification model, it might not be necessary to transform the input at all, i.e. one might solve the downstream on the original input image. Indeed, this is a failure mode we observe in practice — often, the localiser simply learns the identity transform while the classifier learns to classify the non-transformed image. Using more complex classifier architectures makes the STN more prone to this failure mode. This has been observed by other authors [Finnveden et al., 2021], and we investigate the problem in the experiment in Fig. 12. We start by training a differently-sized neural network on MNIST (black, one layer on the x-axis is `[Linear, ReLU, Dropout]`). We compare the performance of this model to (P-)STN models trained on rotated MNIST, test accuracies are plotted in the left panel of the figure. If the localization task is performed perfectly, the (P-)STN models should be able to recover the accuracy on the original, non-rotated dataset. In the right panel, we plot the variance of the (mean) transformations learned by the (P-)STN models. Values close to 0 indicate that the localiser does not transform the image, i.e. it learns the identity transform. Larger values indicate

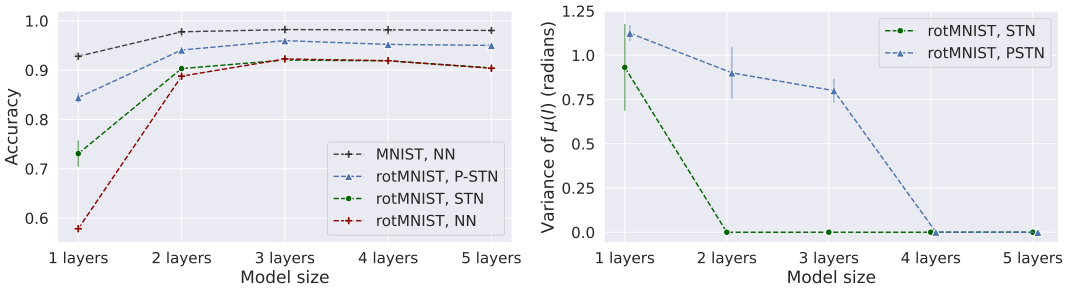


Figure 12: *Left:* Test accuracies for standard NN and (P-)STNs of different depths trained on rotated MNIST, as well as NN baseline on original MNIST (black). The STN (green) model does not usually recover the original images and thus behaves more like a standard NN (red) in most runs. P-STN (blue) un-transforms at least some of the rotations and is closer in accuracy to the NN on original MNIST (black). *Right:* The variance of the learned transformations as a function of model depth. The STN learns the identity for deeper downstream models (this is consistent with the test accuracies we see on the left). P-STN learns to un-transform better, at least when the classifier is simple. For bigger classifiers it predicts the identity transform as well, but performs relatively well nonetheless (see left panel). We report medians ± 1 median absolute deviation over 5 folds.

that the localiser learns transformations. Median results are reported over 5 runs, error bars correspond to one mean absolute deviation. As hypothesized, for larger classifiers the localizers do not transform the images. Due to the increased capacity of the model, we nonetheless achieve decent classification accuracies (left panel). The P-STN learns to localize the rotated images somewhat successfully (large variance in the right panel, and high accuracy on the left) for smaller classifiers. The STN does not localize the images as well, most runs behave like the standard NN on rotMNIST (red), predicting identity transformations only. We conclude that thanks to it 'trying out multiple transformations', the P-STN avoids this failure mode to an extend. We also note that this property, while useful, is somewhat orthogonal to our interest in this work, and we have avoided the failure mode in the experiments of Sec. 5.1 by considering models with *fixed*, pre-trained classifiers.

6 CONCLUSION

We have introduced a probabilistic extension to the spatial transformer network (STN) [Jaderberg et al., 2015]. Our work took motivation from the empirical observation that the STN is often brittle to train as a poorly predicted transformation may prevent the model from getting any gradient signal, resulting in divergent optimization. Our probabilistic STN (P-STN) instead approximates the posterior distribution of transformations using amortized variational inference, and marginalizes accordingly. As is common, marginalization improves the robustness of the model.

Empirically, we, in particular, note two advantages of the probabilistic formulation over the deterministic. First, the performance of the localization network is improved, since the Monte Carlo marginalization effectively amounts to trying many different transformations. Secondly, the probabilistic formulation improves the overall model performance, since the sampled transformations act as data augmentation both during training and during testing. The resulting ensemble of predictions is more accurate and better calibrated than common classifiers as well as the original spatial transformer.

Acknowledgements

MJ is supported by a research grant from the Carlsberg Foundation (CF20-0370). SH was supported by research grants (15334, 42062) from VILLUM FONDEN. This project has also received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement Nr. 757360). This work was funded in part by the Novo Nordisk Foundation through the Center for Basic Machine Learning Research in Life Science (NNF20OC0062606).

References

Henry S Baird. Document image defect models. In *SDIA*, pages 546–556. Springer, 1992.

Gregory Benton, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. Learning invariances in neural networks. *arXiv preprint arXiv:2010.11882*, 2020.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *ArXiv*, abs/1601.00670, 2017.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

Hoang Anh Dau, Eamonn Keogh, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, Yanping, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, Gustavo Batista, and Hexagon-ML. The ucr time series classification archive, October 2018. `https://www.cs.ucr.edu/~eamonn/time_series_data_2018/`.

Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux-effortless bayesian deep learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Nicki Skafte Detlefsen. libcpab. `https://github.com/SkafteNicki/libcpab`, 2018.

Nicki Skafte Detlefsen and Søren Hauberg. Explicit disentanglement of appearance and perspective in generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Nicki Skafte Detlefsen, Oren Freifeld, and Søren Hauberg. Deep diffeomorphic transformer networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4403–4412, June 2018.

Nicki Skafte Detlefsen, Martin Jørgensen, and Søren Hauberg. Reliable training and estimation of variance networks. In *33rd Conference on Neural Information Processing Systems*, 2019.

Christian Ertler, Jerneja Mislej, Tobias Ollmann, Lorenzo Porzi, and Yubin Kuang. Traffic sign detection and classification around the world. *CoRR*, abs/1909.04422, 2019. URL `http://arxiv.org/abs/1909.04422`.

Lukas Finnveden, Ylva Jansson, and Tony Lindeberg. Understanding when spatial transformer networks do not support invariance, and what to do about it. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 3427–3434. IEEE, 2021.

Oren Freifeld, Søren Hauberg, Kayhan Batmanghelich, and John W. Fisher. Transformations based on continuous piecewise-affine velocity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

Ian Goodfellow, Honglak Lee, Quoc V Le, Andrew Saxe, and Andrew Y Ng. Measuring invariances in deep networks. In *NIPS*, pages 646–654, 2009.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org, 2017.

Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schon. Evaluating scalable bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 318–319, 2020.

Søren Hauberg, Oren Freifeld, Anders Boesen Lindbo Larsen, John W. Fisher, and Lars Kai Hansen. Dreaming more data: Class-dependent distributions over diffeomorphisms for learned data augmentation. In *Artificial Intelligence and Statistics*, pages 342–350, 2016.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

Angjoo Kanazawa, David W Jacobs, and Manmohan Chandraker. Warpnet: Weakly supervised matching for single-view reconstruction. In *CVPR*, 2016.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL `http://arxiv.org/abs/1312.6114`.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

Fabian Küppers, Jan Kronenberger, Amirhossein Shantia, and Anselm Haselhoff. Multivariate confidence calibration for object detection. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

Fabian Küppers, Jan Kronenberger, Jonas Schneider, and Anselm Haselhoff. Bayesian confidence calibration for epistemic uncertainty modelling. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, July 2021.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

Yann LeCun, LD Jackel, Leon Bottou, A Brunot, Corinna Cortes, JS Denker, Harris Drucker, I Guyon, UA Muller, Eduard Sackinger, et al. Comparison of learning algorithms for handwritten digit recognition. In *International conference on artificial neural networks*, volume 60, pages 53–60. Perth, Australia, 1995.

Chen-Hsuan Lin and Simon Lucey. Inverse compositional spatial transformer networks. *CoRR*, abs/1612.03897, 2016. URL `http://arxiv.org/abs/1612.03897`.

Gaëlle Loosli, Stéphane Canu, and Léon Bottou. Training invariant support vector machines using selective sampling. *Large scale kernel machines*, pages 301–320, 2007.

David J C MacKay. Model comparison and Occam's razor. *Information Theory, Inference and Learning Algorithms*, pages 343–355, 2003.

David JC MacKay. Bayesian interpolation. *Neural computation*, 4(3):415–447, 1992.

Pola Schwöbel, Martin Jørgensen, Sebastian W Ober, and Mark Van Der Wilk. Last layer marginal likelihood for invariance learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3542–3555. PMLR, 2022.

Ron A Shapira Weber, Matan Eyal, Nicki Skafte, Oren Shriki, and Oren Freifeld. Diffeomorphic temporal alignment nets. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 6570–6581. 2019.

Patrice Y Simard, Dave Steinkraus, and John C Platt. Best practices for convolutional neural networks applied to visual document analysis. In *2013 12th International Conference on Document Analysis and Recognition*, volume 2, pages 958–958. IEEE Computer Society, 2003.

Søren Kaae Sønderby, Casper Kaae Sønderby, Lars Maaløe, and Ole Winther. Recurrent spatial transformer networks. *arXiv preprint arXiv:1509.05329*, 2015.

Andrew Stirn and David A Knowles. Variational variance: Simple and reliable predictive variance parameterization. *arXiv e-prints*, pages arXiv–2006, 2020.

Hiroshi Takahashi, Tomoharu Iwata, Yuki Yamanaka, Masanori Yamada, and Satoshi Yagi. Student-t variational autoencoder for robust density estimation. In *IJCAI*, pages 2696–2702, 2018.

Mark Van der Wilk, Matthias Bauer, ST John, and James Hensman. Learning invariances using the marginal likelihood. In *Advances in Neural Information Processing Systems*, pages 9938–9948, 2018.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Paper C

The Long Arc of Fairness: Formalisations and Ethical Discourse

The Long Arc of Fairness: Formalisations and Ethical Discourse

Pola Schwöbel*
Technical University of Denmark
Copenhagen, Denmark
posc@dtu.dk

Peter Remmers*
Technische Universität Berlin
Berlin, Germany
peter.remmers@tu-berlin.de

ABSTRACT

In recent years, the idea of formalising and modelling fairness for algorithmic decision making (ADM) has advanced to a point of sophisticated specialisation. However, the relations between technical (formalised) and ethical discourse on fairness are not always clear and productive. Arguing for an alternative perspective, we review existing fairness metrics and discuss some common issues. For instance, the fairness of procedures and distributions is often formalised and discussed statically, disregarding both structural preconditions of the status quo and downstream effects of a given intervention. We then introduce *dynamic fairness modelling*, a more comprehensive approach that realigns formal fairness metrics with arguments from the ethical discourse. A dynamic fairness model incorporates (1) ethical goals, (2) formal metrics to quantify decision procedures and outcomes and (3) mid-term or long-term downstream effects. By contextualising these elements of fairness-related processes, dynamic fairness modelling explicates formerly latent ethical aspects and thereby provides a helpful tool to navigate trade-offs between different fairness interventions. To illustrate the framework, we discuss an example application – the current European efforts to increase the number of women on company boards, e .g. via quota solutions – and present early technical work that fits within our framework.

CCS CONCEPTS

• **Computing methodologies → Machine learning**; • **Social and professional topics → Computing / technology policy**.

KEYWORDS

algorithmic fairness, ethics of machine learning, fairness metrics, algorithmic decision making, dynamic fairness modelling

ACM Reference Format:
Pola Schwöbel and Peter Remmers. 2022. The Long Arc of Fairness: Formalisations and Ethical Discourse. In *FAccT '22: ACM Conference on Fairness, Accountability, and Transparency, June 21–24, 2022, Seoul, South Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3531146.3534635

*Both authors contributed equally to this research.

FAccT '22, June 21–24, 2022, Seoul, South Korea

ACM ISBN 978-1-4503-9352-2/22/06…$15.00
https://doi.org/10.1145/3531146.3534635

1 INTRODUCTION

At the core of fair machine learning (fair ML) research lies the question: *What is fairness?* A fundamental goal of research in fair ML is to define ethical standards for ML technologies and to help build tools that live up to such standards. This endeavor has become urgent in light of the rapid advancements within ML which have enabled the widespread use of algorithmic decision making (ADM). The stakes are high, both for society and for individuals, and some of these decision making systems have failed in dramatic and systematic ways: racially biased ADM in the criminal justice system [1], facial recognition failing on women of color [6], sexist hiring [11] and racist search engines [32] are only some notorious examples.

Especially important for auditing the fairness of algorithms are fairness metrics, i.e. formal criteria by which to score fairness, and a multitude of metrics has been proposed. However, despite great research efforts and a plethora of approaches, there are fundamental issues that the field has not, so far, been able to overcome. As noted by Jacobs et al. [24] and Binns [5], some of these issues are consequences of the tendency of fair ML research to conflate formal analysis of fairness with the discussion of ethical principles. While certain basic ideas of fairness are formally constructed as fairness metrics, these formalisms are then analysed too narrowly without entering a (non-formal) *ethical* debate. For example, formal contradictions between two different fairness metrics have been construed as technical flaws of the metrics, when in fact both fairness metrics are perfectly valid formalisations of ethical principles (as is the case for the apparent conflict between individual and group fairness [5], see further discussion in Sec. 2).

In the line of reasoning of this contribution, we argue for a clarification of the roles of formal contributions and ethical debate in fair ML research. Rooted in quantitative fields, fair ML depends on formalisations. However, analysis of formalised criteria alone cannot determine the grounds for a choice between different criteria (other than their formal properties, e.g. whether they are consistent with each other, or whether they have computational properties such as differentiability which make them suitable as a loss function). Fair ML thus also depends on a comprehensive discussion of ethical principles, goals and values. We aim to develop a formalisation strategy that incorporates such ethical considerations, and show that such formalisations can aid the non-formal debates in turn.

By reviewing existing fairness metrics and their weaknesses in Section 2, we identify a second issue in the current fairness debate: Constructed fairness criteria are often not sufficiently contextualised. Procedural fairness criteria assume that an unbiased decision process alone will lead to a fair state of the world (see Sec. 2.1). Operating under this assumption, they neither acknowledge nor adjust for biased data and as a consequence are prone to reinforcing existing inequalities. Outcome based fairness modelling provides a

more promising approach, however, the standard outcome based metrics usually fail to capture ethically relevant structural differences between groups (see Sec. 2.2). That is, they do not investigate *how* a certain outcome distribution arose, or the long-term effects of intervening in such a distribution.

In contrast, we argue that we should optimistically demand of fair ML exactly this: It should be thought of as a tool to *intervene* in the status quo and improve conditions for the previously disadvantaged. The transformation of established bureaucratic procedures towards automation-based processes offers historically unique opportunities for a reevaluation and restructuring of society. Such transformation holds the promise of improving structural conditions for historically disadvantaged groups and individuals via access to, for example, better jobs, wealth and education. In this context, a *fairness intervention* is a procedure that is specifically designed to address and intervene into pre-existing discrimination. A well-known general example for this strategy is the practice of positive (or 'affirmative') action. In contrast to the paradigm of 'blind' decision-making that intentionally excludes certain protected features from the process, respective decisions explicitly take demographic differences into account in order to counteract historic forces of discrimination. We call this motive for fair ML *interventional*.

With this context we will introduce a framework we call *dynamical fairness modelling* in Section 3. Dynamical fairness modelling, we argue, helps bridging the gap between (un-)fairness in the world (the ethical discourse) and formalisations (the formal discourse). It does so because it requires the following steps: It forces the decision maker to explicate their long term *goals* in ethical terms (as opposed to the merely implicit ethical dimension of a predominantly technical choice), their *formalisation* as well as the expected long-term *effects* of any suggested interventions. Rather than evaluating fairness interventions statically and in isolation, dynamic fairness modelling prompts the decision maker to reflect on, model and evaluate the downstream effects of any chosen decision policy – according to the interventional perspective. To illustrate this framework, we will discuss a conceptual as well as a computational example. After reviewing some existing technical work in the direction we propose, we will conclude by returning to a more philosophical treatment in Section 4 where we will also discuss limitations.

2 PROCEDURAL AND OUTCOME-BASED FAIRNESS METRICS

In this section, we briefly present and discuss some technical approaches that are prominent in the debates on fair ML, although we do not claim to give an exhaustive overview. The approaches can be categorised as procedural and outcome-based criteria of fairness, roughly following the distinction between pure procedural justice and perfect procedural justice as introduced by Rawls ([35] p. 74-75). We will demonstrate how discussions of 'static' formal metrics lead to issues that can be addressed by incorporating further context that is initially not present in the existing formalised criteria. We will eventually arrive at a contextualised modelling approach, dynamical fairness modelling, in Section 3.

2.1 Procedural Criteria

Procedural fairness is determined by criteria that refer to the *process* of a decision (as opposed to the *outcome* of a decision). Procedural fairness criteria may follow the ethical principle to treat everyone equally in a decision process, independently of any specific attributes.[1] On the other hand, decision making procedures are *unfair* if they follow principles that are themselves ethically unacceptable, independently of the outcome. Specifically, considering given histories and structures of discrimination, it may be ethically unacceptable to base a decision on certain sensitive attributes, for example using attributes like race or gender in the context of hiring. The so-called 'blindness' approach to anti-discrimination as formalised in the 'fairness through unawareness' criterion constructs a decision procedure that is supposed to be fair by simply not considering any such protected attributes [18].[2] For example, the principle of 'color blindness' refers to racial categories: 'Generally, color blindness minimises the use and significance of racial group membership and suggests that race should not and does not matter.' ([34], p. 200).

Why procedural criteria fail: They neither acknowledge nor adjust for biased data. There are fundamental problems with this approach to anti-discrimination. Although the procedural constraints of the 'blindness' approach might be effective in preventing direct (i.e. explicit) discrimination, other variables can act as proxies for protected attributes. In this case, there is information flowing from the protected attribute A to the outcome even if the category of A is not explicitly used by the model. This happens because structural discrimination is statistically effective in many ways: It correlates protected attributes like race or gender to geographic residence, socioeconomic status, education, medical records, family background, criminal records and other attributes. Consequently, if there is discrimination, its effects are very likely manifest in data. And if the data that correlates to protected attributes is used in a decision making procedure, the process may be as discriminating as if the protected attributes were explicitly used in the first place.

This problem is exacerbated in ML-based ADM. Machine learning works by extracting patterns from large amounts of historical data by statistical inference, referred to as 'training the model', and then using these patterns to determine decisions. Considering this fundamental mechanism by which ML works, it becomes clear that ML can never be better than the data used to train it. We can at best hope that the algorithm perfectly captures the information we have presented it with. But if we train an algorithm on biased training data, it will reflect such biases.

For this reason, the principle of non-discrimination as 'exclusion of protected attributes' is a formal criterion that does neither acknowledge nor adjust biased data. At best, a decision procedure realises equal chances and opportunities for everyone affected. At worst, a procedure mirrors data bias and proliferates discrimination. In this case, a 'blind' decision procedure reproduces a given

[1] According to the ethical goal of 'equal treatment', sufficiently random decisions may be considered fair insofar as probabilities are equal for everyone (cf. [10] for an interesting discussion of the (un-)fairness of random decisions.).

[2] Protected attributes are features such as religious affiliation, age or sexual orientation that are 'protected' by law of many countries. Individuals cannot be discriminated against based on such attributes for example in the context of hiring (e.g. US Civil Rights Act Title VII) or housing (e.g. US Fair Housing Act).

distribution of capabilities and opportunities that was unfair in the first place.

Possible solutions. To deal with the issue of proxies, we might try to somehow 'filter' the data used in the process in a more elaborate way. An example for this approach works by identifying which data (other than the data that explicitly refers to the protected attributes) should or should not be used by a decision making algorithm. For example, Grgić-Hlača et al. [21] propose an approach based on surveying users about whether any feature should be used in a fair decision making process. On the other hand, not all features that correlate with the protected attribute might be unacceptable for ethical decision-making: For example, a job might require the applicant to hold an engineering degree; and holding such a degree is positively correlated with being male in many countries. As a consequence, less women and non-binary individuals[3] might be hired without the gender attribute being used in a discriminatory way. In other words, only causal relations between protected attributes and decision outcomes are problematic in terms of fairness, and only those need to be corrected for. This idea is explored in causal fairness approaches (e.g. [8, 29, 30]). While such approaches provide elegant solutions where causal data is available (i.e. where we know the reason why a certain situation came about), there is a reason why modelling is usually done in the observational sense, based on correlation rather than causation: It is in most cases difficult, if not unfeasible, to model the full causal process leading to a certain feature distribution; e.g. the cultural and historic reasons for women and non-binary individuals not to choose undergraduate degrees in engineering in the first place remain unexamined.

Instead, we can acknowledge present and historic discrimination that result in biased data and work towards ways to address them. A fairness intervention should be thought of as a procedure that is specifically designed to address and intervene into pre-existing discrimination. Corresponding to this *interventional perspective*, we argue for *dynamical fairness modelling* which we will introduce in Sec. 3.

2.2 Outcome-based criteria

The above considerations and further examples from the literature on bias and fairness suggest that we should eventually judge the fairness of the procedure by its outcome. While a certain procedure may seem completely unbiased and non-discriminating by itself, it may appear differently when we look at its outcomes [37]. Perhaps we find out that although a seemingly fair decision procedure carefully precludes sensitive data, it still leads to an apparently unfair distribution of opportunities and goods. Consequently, decision procedures that incorporate potentially biased data should be evaluated by looking at the outcomes. Outcomes can be measured in terms of the distribution of goods, e.g. resources and material goods, but also opportunities, capabilities and well-being. Fairness then correlates to the ethical acceptability of a certain outcome. The ethical goal of a respective fairness intervention could be an equal distribution of goods or, alternatively, a distribution that is proportional to a certain merit. In this setting, an algorithm's fairness can be evaluated by reference to the distribution of outcomes it produces, i.e. the state of a world in which decisions were made according to the algorithm's predictions or recommendations. Generally, outcome-based approaches are suited to bypass the previously mentioned blind spot of procedural fairness, because an evaluation of outcomes is based on criteria of fairness that are to some extent detached from the bias of the original data. For this reason, these approaches seem to be motivated by the idea of controlling potential unfairness by actively neutralizing certain biases (although the interventional stance will turn out to be a more adequate point of view).

Group fairness metrics. Early contributions to algorithmic fairness propose outcome-based criteria such as demographic parity [7] or equality of opportunity [22] (see Table 1 for formalisations of these and other metrics). Applied to the example of hiring and gender, demographic parity corresponds to hiring the same proportion of male and female candidates. Equality of opportunity requires hiring at the same proportions conditioned on the candidates' qualifications. In our example, *qualified* male and female applicants should be hired at the same rates. The difference between demographic parity and equality of opportunity becomes apparent when considering the case of unequal qualification rates between the genders. If indeed more qualified men apply, the latter criteria allows for differences in hiring rates, where the first does not. Because the protected attribute is usually thought to encode the membership to a demographic group (gender, race, etc.), criteria based on such attributes are summarised under the term group fairness.

Individual fairness metrics. Seemingly in contrast to group fairness are so-called individual fairness metrics. According to individual fairness, a decision is fair if similar individuals are treated the same way, or, in terms of Aristotle's account of justice, that similar cases are treated alike. In our example of hiring, to satisfy individual fairness, equally qualified candidates should either both be hired or not hired, regardless of which group they are categorised in. Much effort in technical work on fair ML focuses on evaluating different fairness metrics against each other, and proving various incompatibility statements [9, 13]. Formally, apart from very specific cases, group fairness and individual fairness can not be satisfied simultaneously. If the underlying distribution of features is different between demographic groups, we cannot obtain demographic parity while at the same time treating individuals from both groups the same. In order to achieve demographic parity, we need to allow for preferential treatment of the less qualified group. Binns [5] resolves this conflict by pointing to the shared underlying ethical goal of both individual and group fairness; we briefly discuss his work below.

Why outcome-based criteria fail: They do not acknowledge structural differences between groups. Individual fairness requires a measure for similarity; mathematically speaking, we need a metric to define the distance of individuals x and y in the input space (assume that x belongs to protected group X and y to Y, respectively). In the hiring example, the metric would be defined in terms of some qualification score, and would typically ignore protected attributes when determining similarity. Proceeding in this way, one

[3] Like much of the existing fair ML literature that we build on, we acknowledge the use of overly simplistic categories and false binaries in this work. We view efforts towards inclusive categories and intersectionality as absolutely necessary, and as an orthogonal research direction to static vs. dynamical fairness modelling which is the focus here.

	Fairness principle	Fairness metric (name)	Definition
Fair process	'Blindness': Protected attribute should not be used in the decision.	Fairness through Unawareness	Protected attributes are not explicitly when making predictions [18], $F(X, A) = F(X)$.
	Protected attribute should not cause the decision.	Counterfactual Fairness [30]	$p(F\|do(A = 0)) = p(F\|do(A = 1))$, $do(\cdot)$ is the do-operator which denotes an intervention on the protected attribute.
		(Un-)Resolved Discrimination, Proxy Discrimination [29]	Causal paths between ethically relevant variables and outcome are (un-)blocked, see [29].
Fair outcome	No subjective discrimination: Qualified people should be equally likely to obtain the job/mortgage/etc. across groups.	(Formal) Equal Opportunity [22]	$p(F\|A = 0, Y = 1) = p(F\|A = 1, Y = 1)$
	In addition: *Un*qualified people should also be equally likely to *not* get the job/mortgage/etc. across groups.	Equalised Odds [22]	$p(F\|A = 0, Y = y) = p(F\|A = 1, Y = y)$ for $y \in \{0, 1\}$
	Equal representation, diversity	Demographic Parity	$p(F\|A = 0) = p(F\|A = 1)$
	'Treat like cases alike' *(Aristotle)*	Individual Fairness	$D(F(x_1), F(x_2)) \leq d(x_1, x_2)$ for D and d distance functions in the output and input space.

Table 1: Some fairness principles and their formalisations; for the relationship between fairness principles and ethical goals see Sec. 3. In the right column the notation is as follows. F: the predictor (with slight abuse of notation, this can refer to both a single function as well as a distribution of outcomes, i.e. we do not properly distinguish here between deterministic and probabilistic algorithms), X: the (distribution of) features, A: a protected attribute (e.g. gender or race), Y: the (distribution of) true labels (e.g. whether someone is qualified for the job/mortgage/etc.).

implicitly decides that belonging to group X or Y is ethically irrelevant for the decision at hand, following the principle of 'blindness' as described in Sec. 2.1. But from the interventional perspective, this stipulation is misleading, because we are interested *specifically* in socio-economic, historic and structural differences between groups. Instead of merely ignoring unwanted data that correlates to protected attributes as in the 'blindness' approach, individual fairness should rather construct relevant similarities between selected attributes. A good similarity metric should reflect ethically relevant differences.[4] Interestingly, as Binns [5] shows, when using a similarity metric that accounts for ethically relevant differences between groups, individual and group fairness can become commensurable. Designing such a more holistic similarity metric is not trivial as any choice is necessarily rooted in ethical reasoning and underlying values. Indeed, we need to explicate our ethical stance: 'conflicts are not primarily the result of selecting individual or group fairness measures. Instead, they are likely to be the result of unstated conflicting moral and empirical assumptions regarding the decision-making context' ([5], p. 519).

Metrics like equality of opportunity or equalised odds suffer from a similar shortcoming: They do not account for the different realities of protected groups. The two metrics define unfairness as an unfair distribution of errors, i.e. when opportunities are wrongfully denied for people of certain demographic groups. However, as Eidelson [15] argues, perfectly accurate, i.e. error-free, decisions can be unfair as well if they occur in the context of what he terms *patterned inequality* between groups. As an example, imagine a bank giving out loans. A lending decision is considered accurate whenever the lender can repay. Being wealthy should make it easier to pay back the loan; if the investment does not go as planned, there might be alternative income streams to alleviate the loss and pay back the bank. Thus, an algorithm which only approves loans to wealthy people will be highly accurate, as individuals from this group will likely be qualified in the sense of being able to repay. However, by employing such a decision criterion, people born into less wealthy families will never be afforded the opportunity of taking out a loan to make an investment, say, in their own business, in order to improve the economic situation for themselves. The effect is especially dire in cases where different socio-economic factors are linked (e.g. wealth and race) such that entire communities are systematically excluded from opportunity. Note that this is not a problem specific to machine learning or automation in general, but of merit-based decision making overall. As Kasy and Abebe [28] state: 'under this perspective, inequality [...] is acceptable if it is justified by merit [...], no matter where the inequality [in merit] is coming from'.

Demographic parity seems specifically designed to break such patterned inequalities. It may require drastic positive action, for example approving bank loans at equal rates for men and women. But this can have negative consequences for individuals that belong to the very group that is supposed to benefit, because it ignores the unfortunate reality of the gender pay gap, women's lower wages on average, and thus their potentially lower ability for paying back

[4]For example, in a Rawlsian luck-egalitarian sense, a decision should correct for circumstances negatively affecting an individual's qualification score that lie outside their control.

a loan. Receiving a bank loan that one is unable to repay, however, leads to less financial well-being, a worse credit score and eventually being worse off than without having received the loan in the first place. This, of course, is not to say that the consequences of requiring demographic parity are always negative. More often than not it will be hugely beneficial for an individual to be afforded an opportunity. Nevertheless, the potential harms of group fairness metrics like demographic parity or equalised odds for those groups that are supposed to benefit should be reflected in the implementation of ADM.

To summarise this section, we identify two general approaches to fairness: procedural and outcome-based approaches. Procedural criteria fail to account for existing biases in data and are therefore prone to reproducing existing inequalities. Within the category of outcome-based approaches, we discuss different fairness metrics, formally divided into group fairness metrics and individual fairness metrics. Group fairness metrics entail certain risks for the groups that are supposed to benefit from them. As purely distributive criteria, group fairness metrics neither address nor explicitly control the conditions that lead to a certain distribution. For example, a distribution according to demographic parity is not in itself valuable, but only if it helps to change the social conditions that contribute to the development of strongly disparate distributions in the first place. That means that we should not only discuss fairness in terms of (static) distributions between different groups, but as a result of *processes* that shape and determine these distributions.

Strategy. We conclude that a process-based, i.e. dynamical modelling perspective is necessary to meaningfully reason about fairness in the interventional sense – a perspective that many existing metrics are lacking. We have also seen that existing approaches often suffer from a lack of explicitly stated ethical goals. The necessary ethical debate is sometimes conflated with and obscured by the formal debate, such as in the discourse of the apparent conflict between group and individual fairness metrics. As a consequence, we formulate the following desiderata for fairness modelling:

(1) The ethical goals should be stated explicitly, and independently of formalisation.
(2) Any intervention should be evaluated based on its impact towards ethical goals, i.e. whether it improves the conditions underlying disparate distributions of goods between demographic groups.

The following Section 3 will develop a framework for fairness modelling according to these considerations.

3 DYNAMICAL FAIRNESS MODELLING

We will now outline the implementation of the dynamical fairness modelling framework, first in a short overview (Sec. 3.1), and then with an example. Observing a mainly US-centric debate, we will work with a European case study: gender quotas on company boards as a potential measure to reduce gender inequality in the workforce. This measure has been discussed and/or implemented in multiple European countries such as Norway,[5] Belgium, Italy, France, Germany and the Netherlands [16]; California followed in September 2018 (CA Senate Bill 826, [20]). After this conceptual example (in Sec. 3.2), we will illustrate what a computational implementation of the framework can look like. To do so, we will review existing technical approaches for dynamical fairness modelling, in particular the pivotal 2019 work by Liu et al. [31] (Sec. 3.3 and 3.4).

[5]Norway is not a itself a EU member state, but has re-kindled the positive action debate across the European Union when it introduced a minimum requirement of 40% of women on all company boards of publicly listed companies as early as 2006.

3.1 Implementing Dynamic Fairness Modelling

(1) Explicate Ethical Goals. The first element of our proposed modelling framework is an explication and discussion of the long-term goals in ethical terms, i.e. independent of possible formalisations. While these explications will likely refer to existing philosophical principles of fairness or justice, e.g. to positions like egalitarianism or equality of opportunity, what we call 'ethical goals' is meant to be more concrete and contextualized, especially with respect to the long term effects of any possible intervention. Instead, the explication of a specific ethical goal should refer to a given background of structural discrimination and inequality, ideally by incorporating the specific histories and conditions that are relevant for the context of the projected decision making system. To complement general principles, localized knowledge about racism, sexism, colonialism or classism etc. should play a role in the discussion of ethical goals. Additionally, these reflections should be very specific in terms of those local contexts that will be influenced and transformed by the development of an ADM system.

We give two brief examples for ethical goals here that will be elaborated in the rest of this section. Firstly, consider the realization of the value of diversity in the assembly of teams. A concrete manifestation of 'diversity' will depend on which groups were previously un- or underrepresented and why. For example, when reasoning about women in the workplace, it is useful to consider factors such as the traditionally higher workload for women in the home (see case study in Sec. 3.2). Another example is an institution setting the goal to actively advance substantive equality of opportunity between demographic groups. A good fairness intervention might aim to help those that are structurally disadvantaged due to the local history and culture, not only by affording them opportunities directly but by helping them to successfully compete for those (see example in Sec. 3.3).

(2) Formalisation. In a second step, decision makers approximate a formalisation of the previously explicated ethical goals. In a simple case, this formalisation might simply correspond to one of the existing fairness metrics. For example, as Binns [5] shows convincingly, an egalitarian ethical stance could be formalised in terms of both group or individual fairness metrics. (Formal) equality of opportunity corresponds to the fairness metric of the same name. Ethical goals around diversity and equal representation can mathematically be expressed via the demographic parity metric. The ethical principle to 'treat like cases alike' which is in many contexts required by legislation can be encoded via the individual fairness metric [13]. Table 1 contains some fairness principles and their formalisations; they are discussed in more detail in Sec. 2.

Applying existing fairness metrics in this sense is an easy way to arrive at formalisations of ethical goals; however, they should not always be expected to correlate to existing metrics as easily. As argued in the previous paragraph (1), our ethical goals usually

require a higher level of specificity and contextualisation. In particular, as we will show with an example in the next section, many ethical goals are more robustly formalised under a *long-term* view. This temporal perspective is important to address not only the symptoms of structural discrimination, but also the conditions that produce them. This dimension is not expressed in the standard fairness metrics, which is why we call them 'static'. Under the dynamic modelling point of view, additional formalisations become available. For example, as we will see in Sec. 3.4, Liu et al. [31] suggest optimising for equally distributed features (rather than outcomes) as a proxy for fairness.

(3) Modelling Down-Stream Effects. Once the decision maker has formalised their ethical goal, they can start to evaluate any potential course of action against it. This means developing a mathematical model of the downstream consequences of a given action, e.g. will admitting more female students to university programs increase the number of qualified female applicants for certain positions. Of course, the quality of this model is essential for the success of our approach, i.e. it should be based on empirical research and expert knowledge of the problem at hand. Early technical work on dynamical modelling of algorithmic fairness usually proposes models based purely on assumptions which is also valuable, at least to investigate the framework.

Having broken down the dynamical modelling pipeline, we note that a main advantage lies in its explicitness and, consequently, transparency. Each of the steps corresponds to stating or formalising assumptions in a way that can readily be critiqued and tested. Critiquing the first step corresponds to asking: Do we agree with this ethical goal? Disagreeing about the notion of fairness or justice corresponds to a philosophical debate with multiple stakeholders, and ethicists being domain experts. The second step can be evaluated by asking: Does our formalisation indeed capture the ethical goal we have stated? As [24] points out, such measurement modelling tasks are standard problems in the quantitative social sciences. They can be accomplished by, for example, testing whether the formalisation is consistent in the sense of test–retest reliability: If the same 'fairness-test' comes out differently for very similar scenarios, the operationalisation at question is not robust, a sign of a poor measurement. For the last step we ask: Does a given fairness intervention indeed have the claimed effect? Again, this question can, in principle, be answered with expert knowledge and empirical data whenever the research is available. For example, Kalev et al. [26], survey the effect of a variety of positive action policies on management diversity. If such data is not available yet one might decide to roll out the intervention and measure its effects (given budget and ethical constraints). Of course, there might be cases where we fundamentally cannot know the exact outcomes of a certain intervention in advance. In such cases, we might include our epistemic uncertainty in the model of down-stream effects. In critical applications we might decide on conservative interventions with less potential upsides, but more predictable downstream effects. By enabling us to challenge underlying assumptions and mechanisms dynamical modelling provides an interface for interdisciplinary collaboration between stakeholders, technologists, ethicists, social scientists and other experts.

3.2 An Example: Women on Company Boards

The EU as well as individual member countries have been concerned with gender inequality in the workplace and have discussed and implemented a range of interventions, most notably gender quotas for company boards. Such quotas require the boards of publicly listed companies in the respective countries to contain at least a certain percentage of women, typically between 30 and 40% where such solutions are implemented [25]. This section illustrates dynamical fairness modelling by measures of such a fairness intervention. We note that hiring decisions for company boards are not algorithmic in the sense of being fully automated or processed by machines – certainly, such high stakes personnel decisions are currently made by humans. Rather, they are algorithmic in a broader sense that there is 'a step-by-step procedure for solving a problem or accomplishing some end'[6], i.e. an underlying set of (implicit) rules that the decision makers are following. In this sense, most 'principled' decisions can be considered algorithmic.

(1) Ethical Goals: Equality of Opportunity, Diversity and Representation. The 2013 report on 'Positive Action Measures to Ensure Full Equality in Practice between Men and Women, including on Company Boards' [36] prepared for the European commission identifies three ethical goals (referred to as 'normative goals' in the text) of such interventions. The first goal is to 'improve the ability of the disadvantaged group to compete for the available opportunities', i.e. ensuring substantive equality of opportunity. *Substantive* (or, in Rawls' terms, *fair*) equality of opportunity is distinct from *formal* equality of opportunity, in that it does not require equal hiring criteria on paper, but equality in the *chances to satisfy* those criteria [2]. Secondly, they aim 'to limit the negative effects on women's position in the labour market of the unequal distribution of responsibilities in the family'. The third goal is to 'to ensure the balanced representation of men and women in bodies with significant decision-making powers'. Instead of taking the individual's perspective, this last goal is formulated from society's point of view. It could be interpreted as the value of diversity in itself, via some sort of democratic legitimacy (i.e. bodies of significant decision-making powers should be demographically representative of the people they are governing) or via the improved results achieved by diverse teams [33].

(2) Formalisation: Demographic Parity, but in the Long Term. At a first glance, the third ethical goal seems to translate into a formalisation straightforwardly: 'balanced representation of men and women' corresponds to demographic parity. That is, if the base population consists of 50% women, one would aim for the same proportion of female board members. When comparing with the notion of demographic parity encoded in EU legislation, we note a subtle difference to the classic notion from the fair ML literature where demographic parity is understood to apply to *any single decision in isolation*. However, in this real world example it is usually formulated as a long-term goal, i.e. quotas are to be met within a time frame, typically a small number of years [25]. While this might seem like a political technicality at first (we cannot force companies to fire and hire new boards on the spot), we believe that we see

[6]Definition of algorithm according to https://www.merriam-webster.com/dictionary/algorithm.

a general property of fairness principles and their formalisations at play: They are often best thought of as aspirational long-term goals rather than short-term strategies. Indeed, in our example, people generally agree that more balanced company boards are desirable in the long-term, but disagree on the best measures to achieve such parity: In 2010 [36], '77% of the Europeans are of the opinion that we need more women in management positions [...]. At the same time, Europeans are rather sceptical about strong positive action measures. The Eurobarometer survey found that 44% of European respondents (44% W, 44% M) consider that the most efficient measures consist of encouraging enterprises and public administrations to take measures to foster equality between women and men ("code of good practice") and to fight against stereotypes' while 'concerning the imposition of quotas by law, it is favoured by 19% of European respondents (20% W, 18% M)'.

(3) Modelling Down-Stream Effects: The Effectiveness of Positive Action. Having decided on demographic parity (formalised in the long-term sense) as the ethical goal, states can consider different policies to achieve them. A naive way might be to immediately require demographic parity, i.e. re-appoint company boards in a gender-balanced manner and keep the demographic parity constraint for all future personnel decisions. While fulfilling the criterion on paper, this approach does not seem to actually align with many people's moral intuitions (as seen in the Eurobarometer survey data above). They might disagree with this intervention for the reasons that we are familiar with from the fair ML literature: Assume the reason for seeing few female board members is not blatant sexism, or what economists call taste-based discrimination [4], but rather the applicant pool containing few qualified women according to the current hiring criteria. Then, achieving demographic parity immediately can imply hiring 'less qualified' women, or, given equal qualification, preferring women to men which might seem unfair towards their 'more qualified' male counterparts. Such violation of the equal treatment principle is discussed in the fair ML literature as a contradiction between individual and group fairness (see Sec. 2.2). We note that contrary to this intuition, EU law explicitly allows for preferential treatment in the context of positive action: 'With a view to ensuring full equality in practice between men and women in working life, the principle of equal treatment shall not prevent any Member State from maintaining or adopting measures providing for specific advantages in order to make it easier for the underrepresented sex to pursue a vocational activity or to prevent or compensate for disadvantages in professional careers' (Article 157(4), Consolidated version of the Treaty on the Functioning of the European Union (TFEU)).

Secondly, as we have seen in Sec. 2.2, some argue that drastic preferential treatment might have negative consequences for the women themselves: Women might be perceived as less competent in their jobs when quotas are employed in their selection regardless of their actual qualifications [12]. If they were indeed appointed despite being less qualified, they might be less likely to being re-appointed or recommended by their colleagues for other opportunities. More dramatically for the underrepresented group, under-qualified women in such jobs might lead to statistical discrimination against the group of women as a whole. After observing less qualified female individuals, decision makers might conclude that women in general are less able to perform well in the job. We note that this argument is based on the implicit assumption that there are essentially no qualified females in the candidate pool (since we would still be able to hire the most qualified ones under a quota solution). This seems implausible given the fact that more women than men graduate from higher education programs in the EU: In 2019, 46 % of women aged 30 − 34 had attained tertiary education and only 35 % of men across the EU Eurostat (2021). [7] On the other hand, the 'negative example' argument works in the other direction as well: quotas and the resulting increased representation of women produce more role models and can lead to an increased willingness for women to compete for the jobs [3].

Thirdly, it is not clear that demographic parity is indeed desirable if it is achieved by continuously applying quotas. Fairness interventions are lawful and desirable, but they should tackle the *cause* of the inequalities and should thus be temporary. According to the UN Convention on Elimination of Discrimination against Women (Article 4(1)), positive action measures 'shall be discontinued when the objectives of equality of opportunity and treatment have been achieved' [17]. The goal of a good fairness intervention is that it will become redundant over time.

Interestingly, operationalising demographic parity on boards in this sense illustrates a problem with the ethical goal and its formalisation itself. If we achieve demographic parity by continuing to apply quotas every time we have to make a hiring decision but the actual distribution of qualified candidates never changes, i.e. the parity never becomes the 'natural' state (or the stationary distribution of the process), the strategy does not actually appear to be successful in achieving equality in the workplace. Instead, we want to improve the situation for the underrepresented group and design interventions which actually lead to more women being qualified for those board seats (this might not mean changing the women but changing the qualifications). Thus, the ethical goal in its first formulation above, to 'improve the ability of the disadvantaged group to compete for the available opportunities', turns out to be a more complete picture. Once this is achieved, we can obtain demographic parity without any further interventions because more women will be qualified [8]. The formalised debate has, in a way, informed the ethical one (rather than just vice versa).

Under these considerations, one might define 'robust', long-term demographic parity as goal and develop other, temporary strategies to achieve it by improving women's conditions for competing in the labour market. The range of such alternative strategies is wide: A group of approaches aims to enable mothers to (re-)join the workforce, those include flexible work hours or part-time employment, or providing company childcare facilities. Some countries require nomination parity, i.e. employers have to nominate two candidates, one of each gender for every position [36]. We also might invest more in developing female talent early on, in universities or graduate programs, or invest in diversity training or more inclusive job ads. Amongst those interventions, we naturally prefer those which

[7] Gender statistics. Eurostat. Retrieved from https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Gender_statistics#Education

[8] This is based on the assumption that women and men have similar cognitive markups and would, given 'free' choice choose similar professions in distribution. This is a somewhat controversial assumption (what if women choose not to be on board seats?), but it seems to be consistent with the EU law's conception of gender equality.

are most effective, i.e. the best under our model of down-stream effects. In this example, the first group of measures seems to be the least effective [36]. Wheatley [39] suggests that part- and flexi-time arrangements often have negative effects on women's careers since they are likely to re-enforce the traditional dynamics of women working more in household and families.

This example has illustrated the strengths of the dynamical modelling perspective: As argued theoretically in Sec. 2, we have seen how ethical deployment of any algorithmic decision making system in a complex, real world required its embedding into a much bigger context than what static fairness metrics can provide. If we aim to implement it as a fairness intervention, i.e. under the interventional perspective from Sec. 1, we need to consider any decision's consequences over time, and how those feed back into the features relevant for decisions in the future. The dynamical fairness modelling approach can be a formal language for this. Indeed, it can be helpful for such reasoning by bridging the gap between the ethical and the formal debate. After illustrating the approach conceptually, we will now move on to the technical perspective of how dynamical fairness modelling might be implemented in a computational setting by reviewing early existing work.

3.3 A Computational Example: Liu at al.'s Delayed Impact of Fair Machine Learning

Liu et al. [31] propose a mechanism which allows for temporal analysis of ML decision processes by introducing 'a one-step feedback model of decision-making that exposes how decisions change the underlying population over time'. Under this model, the authors study whether certain fairness criteria indeed improve the of well-being of a disadvantaged group, or whether they might even lead to a decline in the variable of interest. To our knowledge, this is some of the earliest technical work that fits within the framework of dynamical fairness modelling, and we will illustrate here how Liu et al.'s approach is one strategy to implementing it.

Ethical goal: Their ethical goal is to 'promote the long-term well-being of disadvantaged groups' ([31], p. 1).

Formalisation: Two groups A and B associated with a protected attribute are characterised by distributions $\pi_{A/B}$ of qualification scores $\mathcal{X}$. The notion of well-being referred to in the ethical goal is then equated with this qualification score. For example, the authors use an individual's credit score as a proxy for their financial well-being in the lending example. Institutions have selection policies $\tau_{A/B}$ (rates at which score they accept credit applications), and those have down-stream effects on the individuals. In particular, they assume the availability of a function $\Delta : \mathcal{X} \to \mathbb{R}$ that provides the expected change in score for a selected individual at a given score. The expected change for the group as a whole is denoted by $\Delta\mu_{A/B}$. The authors then distinguish between long-term improvement ($\Delta\mu_{A/B} > 0$), stagnation ($\Delta\mu_{A/B} = 0$), and decline ($\Delta\mu_{A/B} < 0$) for the groups A and B. The suggested metric for evaluating a decision making policy refers to the change of this average qualification score. A desirable policy leads to an increased average qualification score for the individuals of the disadvantaged group.

Model of downstream effects: The authors assume access to a function $\Delta : \mathcal{X} \to \mathbb{R}$ that provides the expected change in score for an individual with score x. As discussed in Sec. 3, such a function is in practice difficult to construct. In their lending example, they assume the following simple structure: They denote by $\rho(x)$ the probability of an individual with score x to be able to repay the loan. c_+ is the benefit from being granted a loan and being able to repay, c_- is the cost for the individual of defaulting on the loan (for example, the worsened credit score). Then, $\Delta(x) = c_+\rho(x) + c_-(1 - \rho(x))$.

The authors show that static fairness metrics, especially demographic parity, can under certain conditions lead to a decline of the qualification score, and thus to the protected group being worse off in the long term. This finding ties in with the problems outlined in Sec. 2. As a corrective, the authors suggest optimising for an improvement of the qualification score for the disadvantage group directly rather than applying existing fairness metrics after the fact. This suggestion perfectly aligns with our framework. Instead of deciding on interventions beforehand and evaluating their consequences, we suggest to work backwards from the goal.

3.4 Related Work

Similar in spirit to Liu et al. [31], Zhang et al. [40] discuss the impact of static fairness metrics and constraints on the long term well-being of different demographic groups. Unlike Liu et al.'s work, an individual's qualification is here modelled as a latent, unobservable variable. Observable scores like school grades are viewed as noisy estimates for the underlying qualification – a relevant difference in the light of ongoing debates about discriminating bias of school grades or standardised tests [14, 19]. Their findings again highlight the complexity of the issue: Whether a given, static fairness constraint is beneficial or detrimental downstream depends on the specifics of the problem at hand and cannot be determined without an analysis of the decision's consequences over time.

Kannan et al. [27] analyse fairness policies for college admission and share our view on the essence of the fair ML issue: 'What is often unstated (and perhaps not even explicitly considered by the colleges) is what exactly the long term goals of these policies are, beyond the short term goal of having a diverse freshman class' ([27], p. 2). The authors formalise two such long term goals by analyzing how the college admission and grading policy influences a potential employer's hiring decision. Firstly, downstream equal opportunity requires that suited college graduates are equally likely to be hired independent of their demographic. Secondly, elimination of downstream bias demands that 'rational employers selecting employees from the college population should not make hiring decisions based on group membership' ([27], p. 2). This second criterion is equivalent to demanding that the college grades are distributed such that the employer can apply the 'blindness' criterion from see Sec. 2.2 without obtaining sub-optimal decisions, i.e. hiring candidates with subpar qualifications. Like Zhang et al. [40], this work models a student's true qualification as a latent variable that can only be estimated noisily by standardised test scores or college grades. Their finding consists in yet another 'inconsistency statement': in general, downstream equal opportunity and elimination of downstream bias cannot be achieved simultaneously.

Heidari et al. [23] take a societal perspective rather than focusing on the individual. The authors formalise a mathematical model for allocating opportunities such as college admissions to people. Motivated by an extremely strong correlation between US parents' and their kids' socioeconomic status (thus low intergenerational mobility), the effects of positive action on intergenerational socioeconomic status is analysed. In line with our intuition about dynamic fairness modelling and the importance of a long-term view, the authors find the following: An optimal allocation policy that only takes the current generation into account will not employ positive action. However, when future generations are taken into account the optimal policy – in the utilitarian sense of maximising the number of people who are given an opportunity and succeed – will include positive action. Intuitively, this is because a child of a well-off individual is likely to be well-off themselves, and so giving somebody the chance to improve their socioeconomic status has positive downstream effects for society.

4 CONCLUSION

This work has introduced a framework for dynamical fairness modelling, which we believe to have two main advantages over many of the existing fairness metrics. Firstly, it forces the decision maker to *explicate their ethical goals* and commitments, hereby increasing transparency and helping to disentangle the formal and the ethical debates underlying fair ML. This clarification is motivated by the observation that most problems of fairness cannot be solved in the context of a purely technical discussion. While formalisation and technical implementation of fairness metrics may clarify important aspects, the results remain too limited to address ethical and political issues. Thus, we want to foster a technical debate which is rooted in, and informed by, an ethical one. Secondly, it provides a more *contextualised* approach than existing methods. In particular, it accounts for biased data (as a consequence of inequalities in the status quo) and it provides a better starting point for addressing structural differences between groups, eventually improving the conditions for the previously disadvantaged. We have identified this motive for fair ML as the interventional perspective.

In our thinking about technology's role in the process, we perceive an opportunity. This opportunity, we believe, is not aimed at technological 'solutionism': While a technological approach cannot count as a 'solution' by itself, it can work to suggest a certain level of discourse – specifically, a translation of technical metrics into terms compatible to an ethical assessment (and vice versa). We have seen this interplay of different levels of discourse in Sec. 3.2, where modelling efforts have aided our ethical reasoning. Thus, we propose dynamical fairness modelling as a technically mediated way to present issues of fairness in more appropriate terms.[9]

Limitations. The core of our framework is a model of the downstream effects of any fairness intervention. Developing such a model is difficult. How does a college admission, bank loan or hiring decision today affect an individual's well-being, qualifications and socio-economic status in the future? One might argue that if we had access to such information, we might already be much better at designing fair policies. In the context of ML, this information could come in the form of datasets recording populations over time. Such datasets are not currently part of the standard machine learning toolbox, but could easily be made available given the 'big data' culture and ways we collect large amounts of data on essentially everything. Of course, the issues around privacy and the economy of surveillance practices arising from this type of data collection themselves pose a set of ethical questions.

We note that our modelling approach is relevant to a certain type of decision maker. A somewhat broad scope is required for taking the interventional perspective, both in terms of goals/motivations and competencies. Dynamical fairness modelling is relevant to decision processes that happen on a relatively long timeline, and are aiming to make societal change. Decision makers in a public institution or the government come to mind, and we deem the framework equally relevant from a research perspective. On the other hand, it might be less applicable for actors within companies which structurally are often operating on shorter time horizons, and whose primary goals might be different from changing society. But for those striving to make real change in the 'interventional sense' of improving conditions for the previously disadvantaged, we hope this contribution is useful.

[9] On the mediating role of technology, see [38].

ACKNOWLEDGMENTS

The authors wish to thank Søren Hauberg and his research group for interesting discussions and extensive feedback on this work. Special thanks also to Miguel González Duque for editorial help. The authors declare no additional sources of funding, and no financial interests.

REFERENCES

[1] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. Machine Bias. *ProPublica, May* 23, 2016 (2016), 139–159. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing
[2] Richard Arneson. 2015. Equality of Opportunity. In *The Stanford Encyclopedia of Philosophy* (Summer 2015 ed.), Edward N. Zalta (Ed.). Metaphysics Research Lab, Stanford University.
[3] Loukas Balafoutas and Matthias Sutter. 2012. Affirmative Action Policies Promote Women and Do Not Harm Efficiency in the Laboratory. *Science* 335, 6068 (Feb. 2012), 579–582. https://doi.org/10.1126/science.1211180
[4] Gary S. Becker. 1971. *The Economics of Discrimination.* University of Chicago Press. https://doi.org/10.7208/chicago/9780226041049.001.0001
[5] Reuben Binns. 2020. On the Apparent Conflict between Individual and Group Fairness. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency.* ACM, 514–524. https://doi.org/10.1145/3351095.3372864
[6] Joy Buolamwini and Timnit Gebru. 2018. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Proceedings of the ACM Conference on Fairness, Accountability and Transparency.* PMLR, 77–91.
[7] Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. 2009. Building Classifiers with Independency Constraints. In *2009 IEEE International Conference on Data Mining Workshops.* IEEE, IEEE, 13–18. https://doi.org/10.1109/icdmw.2009.83
[8] Silvia Chiappa and William S. Isaac. 2018. A Causal Bayesian Networks Viewpoint on Fairness. In *IFIP International Summer School on Privacy and Identity Management.* Springer, 3–20.
[9] Alexandra Chouldechova. 2017. Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. *Big Data* 5, 2 (June 2017), 153–163. https://doi.org/10.1089/big.2016.0047
[10] Kathleen Creel and Deborah Hellman. 2021. The Algorithmic Leviathan: Arbitrariness, Fairness, and Opportunity in Algorithmic Decision Making Systems. *Virginia Public Law and Legal Theory Research Paper* 2021-13 (2021).
[11] Jeffrey Dastin. 2018. Amazon scraps secret AI recruiting tool that showed bias against women. https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G
[12] Jacquelyn S. DeMatteo, Gregory H. Dobbins, Stephanie D. Myers, and Carolyn L. Facteau. 1996. Evaluations of Leadership in Preferential and Merit-Based Leader

Selection Situations. *The Leadership Quarterly* 7, 1 (March 1996), 41–62. https://doi.org/10.1016/s1048-9843(96)90034-x

[13] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness Through Awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference on - ITCS '12*. ACM Press, 214–226. https://doi.org/10.1145/2090236.2090255

[14] Joan L. Eberle and Gary L. Peltier. 1989. Is the SAT Biased? A Review of Research. *American Secondary Education* (1989), 17–24.

[15] Benjamin Eidelson. 2021. Patterned Inequality, Compounding Injustice, and Algorithmic Prediction. *American Journal of Law and Equality* 1 (Sept. 2021), 252–276. https://doi.org/10.1162/ajle_a_00017

[16] Annette Ekin and Florence Villesèche. 2018. Quotas Get More Women on Boards and Stir Change from Within. https://ec.europa.eu/research-and-innovation/en/horizon-magazine/quotas-get-more-women-boards-and-stir-change-within

[17] Marsha A. Freeman, Christine Chinkin, and Beate Rudolf. 2012. *The UN Convention on the Elimination of All Forms of Discrimination Against Women: A Commentary*. OUP Oxford.

[18] Pratik Gajane and Mykola Pechenizkiy. 2017. On Formalizing Fairness in Prediction with Machine Learning. *arXiv preprint arXiv:1710.03184* (2017).

[19] Saul Geiser. 2020. SAT/ACT Scores, High-School GPA, and the Problem of Omitted Variable Bias: Why the UC Taskforce's Findings Are Spurious. Research & Occasional Paper Series: Cshe. 1.2020. *Center for Studies in Higher Education* (2020).

[20] Marina Gertsberg, Johanna Mollerstrom, and Michaela Pagel. 2021. *Gender Quotas and Support for Women in Board Elections*. Technical Report. National Bureau of Economic Research. https://doi.org/10.3386/w28463

[21] Nina Grgić-Hlača, Muhammad Bilal Zafar, Krishna P. Gummadi, and Adrian Weller. 2018. Beyond Distributive Fairness in Algorithmic Decision Making: Feature Selection for Procedurally Fair Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[22] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of Opportunity in Supervised Learning. *Advances in Neural Information Processing Systems* 29 (2016), 3315–3323.

[23] Hoda Heidari and Jon Kleinberg. 2021. Allocating Opportunities in a Dynamic Model of Intergenerational Mobility. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*. ACM, 15–25. https://doi.org/10.1145/3442188.3445867

[24] Abigail Z. Jacobs and Hanna Wallach. 2021. Measurement and Fairness. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*. ACM, 375–385. https://doi.org/10.1145/3442188.3445901

[25] Vera Jourova. 2016. Gender Balance on Corporate Boards: Europe is Cracking the Glass Ceiling. *Brussels: European Commission* (2016).

[26] Alexandra Kalev, Frank Dobbin, and Erin Kelly. 2006. Best Practices or Best Guesses? Assessing the Efficacy of Corporate Affirmative Action and Diversity Policies. *Am Sociol Rev* 71, 4 (Aug. 2006), 589–617. https://doi.org/10.1177/000312240607100404

[27] Sampath Kannan, Aaron Roth, and Juba Ziani. 2019. Downstream Effects of Affirmative Action. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*. ACM, 240–248. https://doi.org/10.1145/3287560.3287578

[28] Maximilian Kasy and Rediet Abebe. 2021. Fairness, Equality, and Power in Algorithmic Decision-Making. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*. ACM, 576–586. https://doi.org/10.1145/3442188.3445919

[29] Niki Kilbertus, Mateo Rojas-Carulla, Giambattista Parascandolo, Moritz Hardt, Dominik Janzing, and Bernhard Schölkopf. 2017. Avoiding Discrimination Through Causal Reasoning. *arXiv preprint arXiv:1706.02744* (2017).

[30] Matt J. Kusner, Joshua R. Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual Fairness. *arXiv preprint arXiv:1703.06856* (2017).

[31] Lydia T. Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. 2019. Delayed Impact of Fair Machine Learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. PMLR, International Joint Conferences on Artificial Intelligence Organization, 3150–3158. https://doi.org/10.24963/ijcai.2019/862

[32] Safiya Umoja Noble. 2018. *Algorithms of Oppression*. NYU Press. https://doi.org/10.2307/j.ctt1pwt9w5

[33] Scott E. Page. 2019. *The Diversity Bonus*. Princeton University Press. https://doi.org/10.2307/j.ctvc77fcq

[34] Victoria C. Plaut, Kecia M. Thomas, Kyneshawau Hurd, and Celina A. Romano. 2018. Do Color Blindness and Multiculturalism Remedy or Foster Discrimination and Racism? *Curr Dir Psychol Sci* 27, 3 (May 2018), 200–206. https://doi.org/10.1177/0963721418766068

[35] John Rawls. 2009. *A Theory of Justice*. Harvard University Press. https://doi.org/10.2307/j.ctvkjb25m

[36] Goran Selanec and Linda Senden. 2013. *Positive Action Measures to Ensure Full Equality in Practice between Men and Women, including on Company Boards*. EUR-OP.

[37] Stefan T. Trautmann and Gijs van de Kuilen. 2016. Process Fairness, Outcome Fairness, and Dynamic Consistency: Experimental Evidence for Risk and Ambiguity. *J Risk Uncertain* 53, 2-3 (Dec. 2016), 75–88. https://doi.org/10.1007/s11166-016-9249-4

[38] Peter-Paul Verbeek. 2011. *Moralizing Technology*. University of Chicago Press. https://doi.org/10.7208/chicago/9780226852904.001.0001

[39] Daniel Wheatley. 2016. Employee Satisfaction and use of Flexible Working Arrangements. *Work, Employment and Society* 31, 4 (April 2016), 567–585. https://doi.org/10.1177/0950017016631447

[40] Xueru Zhang, Ruibo Tu, Yang Liu, Mingyan Liu, Hedvig Kjellström, Kun Zhang, and Cheng Zhang. 2020. How do Fair Decisions Fare in Long-Term Qualification? *arXiv preprint arXiv:2010.11300* (2020).