Quantum Machine Learning in a World of Uncertainty

Towards Practical Applications with Quantum Neural Networks

Foldager, Jonathan

Publication date:
2023

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

Technical University of Denmark

DOCTORAL THESIS

Quantum Machine Learning in a World of Uncertainty

Towards Practical Applications with Quantum Neural Networks

Author:

Jonathan Foldager

Supervisors:

Professor Lars Kai Hansen

Professor Ulrik Lund Andersen

Professor Jan Madsen

A thesis submitted in partial fulfillment of the requirements for the degree Doctor of Philosophy (Ph.D.)

Submitted April 15, 2023

DTU Compute

Department of Applied Mathematics and Computer Science

Foreword

This thesis was carried out at the Section for Cognitive Systems (CogSys), part of the Department of Applied Mathematics and Computer Science (DTU Compute) for the degree of Doctor of Philosophy (PhD/DPhil) at The Technical University of Denmark (DTU). Generously funded by the William Demant Foundation (previously Oticon foundation) in combination with a travel grant from Stibo foundation the projected spanned around 3.5 years where 3 months was spent on the University of Oxford and the remaining time was spent physically at DTU or from home during the COVID-19 pandemic. Main supervision was done by Professor and Head of Section Lars Kai Hansen and co-supervision was done by Professor and and Head of Section Ulrik Lund Andersen together with Professor and Head of Department Jan Madsen.

A total of four scientific contributions is included in this thesis: one published in Scientific Reports — Nature, one currently under review at New Journal of Physics, one under review at the Uncertainty in Artificial Intelligence (UAI) 2023 conference and one preprint. These papers are throughout this thesis referred to as Paper A, B, C and D each having a dedicated chapter 4, 5, 6 and 7, respectively:

A **J. Foldager**, A. Pesah, and L.K. Hansen. Noise-assisted variational quantum thermalization. *Scientific reports*, 12(1):1–11, 2022 [1] (Chapter 4)

B **J. Foldager** and B. Koczor. Can shallow quantum circuits scramble local noise into global white noise? *arXiv preprint arXiv:2302.00881*, 2023 [2] (Chapter 5)

C **J. Foldager**. Actively learning quantum machine learning architectures from related problems. 2023 [3] (Chapter 6)

D **J. Foldager**, M. Jordahn, L.K. Hansen, and M.R. Andersen. On the role of model uncertainties in bayesian optimization. *arXiv preprint arXiv:2301.05983*, 2023 [4] (Chapter 7)

This thesis not only aims at connecting the relatively wide-spanning results of the papers; it also contains a more general introduction and motivation aimed at being readable to an even broader audience before diving into the mathematics and results.

On the structure and notation of this thesis

This thesis is aimed at readers both in the machine learning and quantum computing communities. While the two communities are very different in many aspects, they also share a lot of the underlying mathematics involved (e.g., linear algebra), however with different notation. Coming from the machine learning side of things, I am convinced it should be possible to meaningfully explain at least some parts of quantum mechanics to a broader audience without having to take an undergraduate degree in physics. This thesis therefore aims at connecting the two areas, machine learning and quantum computing; something useful to both computer scientists and theoretical physicists. Also, I like telling stories and this thesis is the perfect excuse to tell one; a story I wish there was written when I started my Ph.D. Thus, careful considerations went into how the background theory is introduced, and as a consequence the reader might notice occasional "double"-notation in some equations. For example,

$$\langle \phi | \psi \rangle = c, \tag{1a}$$

$$\boldsymbol{\phi}^H \boldsymbol{\psi} = c. \tag{1b}$$

will throughout this thesis refer to "a" = "quantum"-notation and "b" = "machine learning"-notation. Although the choice aiming at a wider audience results in a longer thesis, this effort hopefully pays of for more readers.

Thank you for reading and I hope you enjoy my work.

- Jonathan

Acknowledgements

Many who have obtained a Ph.D. describe it as an adventure with ups, downs, plenty of action, tons of unpredictability, fun and life changing events. After having been through one myself, I can confirm that this was indeed applicable to my project as well. These next sentences constitutes my endless gratitude towards the people who made it possible to start, continue and finish this journey. These fantastic people put their trust in me, guided and assisted me and in most cases taking a huge leap of faith; they truly deserve some glory so here we go.

My first words of gratitude goes to the William Demant Foundation (previously Oticon Foundation) for generously giving me one of their highly competitive Ph.D. scholarships. Reading back on our original project proposal which contained highly ambitious ideas, they indeed took a chance on me. Throughout the entire project they have been remarkably flexible and understanding of my various request including a temporary transition to part-time in order to start my own consultancy business, accepting leaves of absence for teaching and a three months internship at Apple. I am endlessly grateful that Danish companies such as Oticon/William Demant invest in Ph.D. projects, even when projects are bold and perhaps far from direct applications.

My deepest thankfulness goes to my main supervisor Professor Lars Kai Hansen who I have had the tremendous honour of being one of the few lucky ones to be so directly supervised and influenced by. I thank Lars for his continuous support, his caring leadership, his ability to always approach me with a smile, his full attention and patience. I also thank my co-supervisors Professors Ulrik Lund Andersen and Jan Madsen for their commitment to the project and for always being enthusiastic when I presented my work to them. When starting my Ph.D. back in the summer 2019 I reached out to one of the world's leading scientists in quantum machine learning: Professor Peter Wittek. He quickly circled back to me and ended up inviting me to University of Toronto to meet in person and kick start my Ph.D. with a remote research collaboration. Everything looked promising and I truly enjoyed those days. Shortly after I returned home, I was devastated to learn Peter was lost in an avalanche during an expedition in

the Himalayian mountains. Peter helped me a lot without even knowing me, and for that I am deeply thankful. Let us spare a thought for him. So too I thank to Professor Michael Kastoryano who took me under his wing in the first year of my Ph.D. in a time where I knew very little about what I was doing. Again at the very end of my Ph.D., Michael agreed to discuss research projects with me and I am very thankful to getting to work with him. My gratitude is also extended to Professor Michael Riis Andersen for his guidance and substantial assistance in our work regarding uncertainty in Bayesian Optimization. I am very glad for having worked with you and the work we did together with Ph.D. student Mikkel Jordahn whom I also want to thank for his hard work and taking the paper the last important steps. I thank my partner-in-crime Ph.D. student Arthur Pesah at UCL for the work we did on the "Noise-Assisted variational quantum thermalizer" paper which led to my first publication as a Ph.D. student. We had a ton of fun making that paper. I owe huge thanks to Dr. Bálint Koczor and Professor Simon Benjamin for hosting my external stay in Oxford, both being so helpful throughout the project. It was an amazing stay, I learned so much, and everyone was so friendly and inclusive. Thanks! In the winter of 2022 I was lucky enough to get an internship at Apple in California as a machine learning intern. I want to thank both Jacob Vestergaard, Gautam Muralidhar and the rest of the team for putting their faith in me and providing me with a perfect combination of independence and guidance as well as offering me a full-time job starting in October 2023. I accepted their generous offer and cannot wait to get started.

Despite large chunks of my project was carried out from home due to the COVID pandemic, I did get to spend a significant time with my office mates at DTU. A special thanks goes to Rasmus Høegh, David Frich, Jesper Løve, Alma Lindborg and Rui Liu for the countless discussions on machine learning, neuroscience, quantum computing, politics and everything in between; constantly forcing me to formulate quantum physics to computer scientists as well as taking a stance on the latest news in politics. A big thanks also goes to Alexander Neergaard Zahid for always being up for a cup of coffee and a conversation about rock music, fermented foods or expensive wines that we will never be able to afford. So too I thank Maxim Khomiakov for the many rounds of golf we played, which undoubtedly helped taking my mind away from being in the "Ph.D.-bubble" for too long at a time. I also want to thank Secretary/administrative Coordinator at CogSys Anne Ringsted for her assistance with all the administrative obligations regarding external stay and handing in this thesis. I also owe thanks to Professor Morten Mørup for helping me out with last-minute writing recommendation letters as well as countless discussions on machine learning theory.

Last but so far from least, I want to express my infinite gratitude to my family and friends without

whom it would have been a colorless three plus years — and I say that as a red/green colorblind person. Without hesitation, my deepest thanks must be given to my fiancée Elisabeth for her love and support. Thank you for always being there for me; without you, nothing like this was possible.

English Popular Science Summary

When computers learn to recognize, predict and generate patterns from data without explicit programming we call it machine learning (ML). Although we are starting to see the first glimpses of ML becoming an integral part of society, there exists many mathematical problems requiring extensive computational resources. Meanwhile, Quantum Computers (QCs), which holds great promise for speedups and increased capacity, are being built with the ambition of revolutionizing specific areas of the information technology industry; ML being one of the "killer applications". While building large, scalable and fault-tolerant quantum computers is a difficult job likely to finish decades from now, we already today have access to noisy intermediate-scale quantum (NISQ) computers. In this exciting NISQ-era, the interface between QC and ML—referred to as quantum machine learning (QML)—has been born as a rich and fast-moving research field which studies how a QC and ML can assist each other.

Despite practical applications of current QML is widely debated, at least three core questions are considered important regardless of current practicality. The first question is, if there are useful tasks NISQ computers which can be used to accelerate ML. The second question is, given the noisy nature of NISQ hardware can one find key characteristics in the accumulated noise that leads us closer to practical applications. And lastly, can we expand our ML toolbox working together with the quantum computer in order to solve more general problem classes in QML.

In order to approach these questions, this thesis offers four scientific papers aiming at delivering results that gets us closer to relevant answers. The first paper develops a quantum algorithm that approximates a specific quantum state which can be used in one of the most computationally hard ML tasks: sampling from complicated high dimensional probability distributions. The second contribution proposes metrics that evaluate noise characteristics in NISQ hardware such that algorithms running on the hardware can be paired with appropriate error mitigation strategies which is paramount to achieve practical NISQ protocols. The third article develops a meta-learning protocol exploiting similarity between key problems in quantum physics and thus enables one to learn across problems in minimum

energy eigenstate estimation; a problem very relevant to machine learning as well. The fourth and last paper studies how uncertainty calibration affects the aforementioned meta-learner, and hence this thesis also provides ML investigation with applicability in QML.

In the end, the thesis aims both at drawing connections between all contributions as well as giving an introduction to QML for computer scientists. While the marriage of NISQ and ML is still bumpy and their combined future is hard to predict, we end on a positive and optimistic note on the progress of QML in the near term.

Dansk Populærvidenskabelig Opsummering

Når computere lærer at genkende, forudsige og generere mønstre fra data uden at være eksplicit programmeret, kalder vi det maskinlæring (ML). Selvom vi begynder at se de første glimt af, at ML bliver en integreret del af samfundet, er der stadig mange matematiske problemer, der kræver store beregningsmæssige ressourcer. I mellemtiden bliver kvantecomputere (QCs), som har stor potentiale for hastighedsforbedringer og øget kapacitet, bygget i håb om at revolutionere bestemte områder af informationsteknologien, herunder ML som anses for en "killer application". Mens det at bygge store, skalerbare og fejltolerante kvantecomputere er en vanskelig opgave, som sandsynligvis vil tage årtier, har vi allerede i dag adgang til noisy intermediate-scale kvantecomputere (NISQ). I denne spændende æra er grænsefladen mellem QC og ML, også kendt som kvantemaskinlæring (QML), blevet født som et rigt og hurtigt-bevægende forskningsfelt, som undersøger, hvordan en QC og ML kan assistere hinanden.

Selvom de praktiske anvendelser af den nuværende QML debatteres, betragtes mindst tre kernespørgsmål som vigtige; uanset den nuværende praktiske anvendelighed. Det første spørgsmål er, om der er nyttige opgaver for NISQ, som kan bruges til at accelerere ML. Det andet spørgsmål er, om man—givet den støjende karakter af NISQ-hardware—kan finde nøglekarakteristika i den akkumulerede støj, der fører os tættere på praktiske anvendelser. Og sidst, men ikke mindst, kan vi udvide vores ML-værktøjskasse og således assistere kvantecomputeren til at løse mere generelle klasser af problemer i QML.

For at kunne besvare disse spørgsmål, tilbyder denne afhandling fire videnskabelige artikler med formål om at levere resultater, der bringer os tættere på relevante svar. Den første artikel udvikler en kvante-algoritme, der bringer computerens kvantebits i en specifik tilstand, som kan anvendes i en af de mest beregningsmæssigt krævende ML-opgaver: at tage stikprøver fra komplicerede højdimensionelle sandsynlighedsfordelinger. Den anden artikel foreslår metrikker, der evaluerer støjegenskaber i NISQ-hardware, således algoritmer, der kører på hardwaren, parres med passende fejlreduceringsstrategier, hvilket er afgørende for at opnå praktiske NISQ-protokoller. Artikel nr. 3 udvikler en meta-

læringsprotokol, der udnytter ligheder mellem problemer i kvantefysik og dermed muliggør læring på tværs af problemer i estimeringer af minimums energi-egentilstande; et problem, der er meget relevant i generel mønstergenkendelse. Den fjerde og sidste artikel undersøger, hvordan usikkerhedskalibrering påvirker den førnævnte meta-læringsalgoritme, og dermed giver denne afhandling også en undersøgelse af klassisk ML metoder til anvendelse i QML.

Afhandlingen sigter både mod at illustrere forbindelserne mellem alle fire bidrag samt give en dybdegående introduktion til QML for ingeniøerer uden en fysik uddanelse. Selvom ægteskabet mellem NISQ og ML stadig er uvist, og deres kombinerede fremtid er svær at forudsige, slutter afhandlingen i et optimistisk udgangspunkt om fremskridtene i den nærtstående fremtid inden for QML.

Contents

Chapter 1

Introduction

THE interplay between Machine Learning (ML) and Quantum Computing (QC) are at the same time exciting, mind-blowing, very difficult to master, and holds great potential for high impact applications [5]. While we have already witnessed the first glimpses of ML products in several fields [6] perhaps large language models [7, 8] currently being the most debated examples (Chat-GPT [9, 10] going viral [11, 12]), quantum computers are still in their early development phase [13]. Quantum processors are currently small, noisy, and they have little practical applicability [14]. However, a lot of work has gone into the theory of quantum computation [15], that is, what algorithms we—at least in theory—should be able to run with this new type of hardware. The key difference between normal (classical) computers and quantum computers lies in *how* the information is stored and processed. Their fundamental difference ties all the way back to how electrons, atoms and molecule behave; with properties such as superposition, interference and entanglement. In this chapter, we will informally introduce the wonders of quantum physics by taking us back to the physical realization of a computer and the chapter will end with how ML fits into the picture.

Classical computers store information (e.g., text, images, movies) is *bits*, which are sequences of zeros and ones; on's and off's. On's and off's of what? It does not matter but it could be light bulbs. Can we store information in light bulbs? Yes! As long as we have enough light bulbs, and we agree on a system of how to store (encode) and read (decode) information, we can store any information we want. For example, we can make a small calculator using, say, three light bulbs. Each of them can be either zero (lights off) or one (light on), and let us picture the three light bulbs being collected them the following way: "[third, second, first]". That is, "$[1, 1, 0]$" means that light bulb three and two are on but light bulb one is turned off. We can also refer to each light bulb as one *bit* of information since the

smallest unit of information we can think of is if something is or is not. Let us agree that the number zero is encoded as all light bulbs are turned off, that is $[0, 0, 0] = 0$. The number one, we can encode as the first bulb being on and the two others turned off: $[0, 0, 1] = 1$. Further more, $[0, 1, 0] = 2$, $[0, 1, 1] = 3$, $[1, 0, 0] = 4$, $[1, 0, 1] = 5$, $[1, 1, 0] = 6$, $[1, 1, 1] = 7$. In total, our collection of three light bulbs can be in 1-of-8 states, since each light bulb can be on or off giving $2 \cdot 2 \cdot 2 = 8$ states, or more general 2^N states for N bits. It does not take crazy imagination to see that having more light bulbs (N bigger than 3), we could store numbers over 7, or even letters, music and movies. As long as we agree on a system to encode the information into the light bulbs and how to decode the light bulbs into pixel values on a screen or how speakers should vibrate. Our three bits can be in one and only one of the 8 states at the time and by turning one or more light bulps on or off, we can change the *overall* state of all the light bulbs; a concept we will use to make our calculator. Let us for example add two the numbers [0,0,1] (i.e. one) and [1,0,1] (five). For this we need an operation, call it O_add, that takes two 3-bit numbers as input and outputs the adding result a three-bit number, i.e. $O_{add}([0, 0, 1], [1, 0, 1]) = [1, 1, 0]$. We thus need some operation that leaves the first light off but turns on the second and third light bulbs. Physically storing information in light bulbs is a waste of space and time, so in practice uses something smaller than a light bulb but have the same properties of being able to be on or off: transistors. Each transistor is one *bit*. For now, it still is abstract how one would build such the O_add operator, but if we can build this theoretical idea into something physical, for example using transistors to store the information, smart wiring to operate on the transistors, buttons so we humans can control what operations to apply to which bits, and a screen to display the result, we have a computer! Our smartphones, laptops, cars, planes, ovens, and bike computers are all made up from this idea of storing and processing information, and look how far it gotten us; do we even need *fundamentally* new computing technology?

A quantum computer is fundamentally different than a classical computer. It does not store information in light bulbs, transistors or anything that only has two states. As a consequence of quantum physics a quantum computer stores information in objects called quantum bits (qubits) which can be in *infinitely* many states. Most importantly, classical computers do not take computational advantage of the laws of quantum mechanics, but qubits do. And the results are astonishing, daunting and almost magical which has led to future predictions of quantum computing to have profound consequences for society [16]. To get a grasp of the potential power of QC, it is useful to introduce the concept of *computational time complexity classes*: an important part of computer science that aims to categorize problems into how much computational time their best known implementation takes to solve, worst case [17]. As an

example, we can focus on a specific problem: given a number k of size N find the prime numbers q and p that satisfy $k = p \cdot q$. For example, if k is 161, an $N = 3$ digit number, the solution is $q = 7$ and $p = 23$. This problem is known as the "prime factorization problem" and it is a hard problem to solve for large digit numbers, i.e., when N is big [18] since we worst case have to check all combinations up to the square root of N. It therefore acts as a cornerstone in so-called Rivest-Shamir-Adleman (RSA) encryption schemes that banks use to protect sensitive information [19]. The best known factoring algorithm [20] (running on normal/classical hardware) scales sub-exponentially in time with the number size N which means that for a, for example, $N = 600$ digit number it takes on the order of the age of the universe to find p and q. At least for factoring algorithms breaking bank encryption, it looks like our pension are safe, for now. There exists several time complexity classes, but here are a few examples of time scales for what it would take if the prime factorization problem had another time complexity than sub-exponential. For $N = 600$ (assuming each run takes one the order of a second):

- Logarithmic time: order of a few seconds.
- Linear time: order of a few minutes.
- Polynomial time: order of a few years.
- Sub-exponential time: order of billions of years.
- Exponential time: order of $\approx 10^{70}$ universe lifetimes.

Compared to the age of the universe, polynomial time (a few years at worst) does not seem so bad. Finding a new algorithm that turns the factoring problem into a polynomial time problem could have huge consequences to society. So when Shor in 1994 came up with a quantum computing algorithm that exactly did this [21], it surprised many. Shor's algorithm showed that, at least in theory, factoring could be done in polynomial time; much faster than any known algorithm up until then. Although exciting QC results existed prior to Shor's paper [22, 23], it can be argued that Shor's factoring algorithm was a tipping point where much "quantum excitement" (as well as fear) started—at least among some physicists [24].

Already in 1985, Deutch conjectured that a quantum computer should be able do things beyond what classical ones could ever do [25], and in 1992 he and Jozsa published an algorithm demonstrating exponential speedups [26, 27] over classical counterparts, albeit with little known practical usage [28, 29]. Following Shor's prime factorization paper in 1994, came many exciting theoretical results,

one being Grover's database search algorithm [30] promising database search quadratically faster than classical counterparts [31]. Moreover, Lloyd proved quantum computational universality [32] which, up until then, was a Feynmann conjecture from 1982 [33] stating that a quantum computer could be used to fully simulate any local quantum system[1]. Fast forward to today, an entire Zoo of quantum algorithms exists with corresponding statements about the speedup over classical counterparts [34]. The Zoo also include quantum simulating (which we have not yet defined what is) algorithms; a hard problem to do for a normal/classical computer. To this date, simulating quantum systems is closely linked to state of the art ideas in quantum machine learning (QML) and this will be one of the major focuses of this thesis.

In these years of increasing quantum excitement, some resistance and scepticism was present [35]. Given that quantum computers need quantum mechanical objects such as electrons (which we know are extremely sensitive to interactions with the surroundings) to store information and do computations, how can one ever make a scalable computer without almost immediately loosing the stored information [36]? Inspired by ideas from classical computing, a major breakthrough, also co-developed by Shor in 1995, came to be: quantum error correction (QEC) [37, 38]. Shor et al. showed that even if some of the information in the subatomic particles was lost to surroundings it was still possible to do net error-free quantum computation as long as the time factor it takes before information is lost (also called decoherence time), was not too short [39]. Together with the threshold theorem [40, 41, 42], the combined theoretical results meant that it all came down to engineering the hardware such that it had enough *physical* qubits having long enough decoherence time to make up a sufficient amount of *logical* qubits which has acceptable error rate relative to the problem one tries to solve. Although current state of the art quantum hardware only houses a few hundred noisy qubits, ambitions are high in many of the established information technology companies [43, 44] as well as several startups [45, 46]. Putting the state of current quantum technology in perspective with latest estimates on the number of qubits required to run Shor's factoring algorithm (20 million physical qubits [47]) and thereby break RSA encryption in a few hours, it might seem like a very long journey before quantum computers become a real thread/asset to society. And indeed, the time scale of if/when quantum computing starts having an impact is being debated, from a few years to decades [48] to "if ever" [49].

But we already have quantum computers today; they exist! Although these are noisy intermediate-scale quantum (NISQ) computers, a term recently coined by Preskill [50], there might be useful infor-

[1] A paper with one of my favorite ways to start a conference article: "On the program it says this is a keynote speech–and I don't know what a keynote speech is" - Richard Feynmann.

mation or applications to gain from studying them. This is exactly the subject of this thesis: to assume that the quantum processor is small and noisy, and then ask what can we do or learn from this in relation to machine learning. If one feels that quantum computing is in itself difficult to master, it can be meaningfully argued that NISQ algorithms are even harder, the main reason being the need for knowledge of other disciplines such as statistical mechanics, chemistry, optimization, and not forgetting quantum information theory. In order to get there, we will take the following approach: we will introduce some history and background of the quantum physical experiments performed in the twentieth century, and subsequently lay out the language and tools necessary to build a quantum computer. Chapter 2 will highlight the similarities, differences and potential interplay between NISQ computers and machine learning, hopefully bringing it all together such that the motivation behind the scientific contributions is solid and provides a meaningful overall story. Because, what is "quantum"? And how does machine learning fit into the story? In the next section, we introduce superposition (Section 1.1.1), interference (Section 1.1.2), and entanglement (Section 1.1.3), followed by sections on how to build a quantum computer (Section 1.2) and what quantum machine learning is (Section 1.3), before we formalize everything in Chapter 2.

1.1 Predicting the Outcome of Experiments

Physics is about finding and predicting patterns from observations [51]. The same can be said about machine learning [52]. Does that mean physics = machine learning? No, but in many aspects, they indeed share the same idea of using data and mathematical models to make predictions about the real world. Maybe that also explains why many physicists find the methods and approaches interesting and some are prominent in the research field. A key dimension the two subjects have in common is their immediate closeness to the scientific method [53], by means of collecting and analyzing data that is used to infer general behavior. But there are also key differences between the two subjects. For example, physics aims at not only predicting observations, but also giving logical explanations using the mathematics, however complicated and counter intuitive the math might be. To put it differently: *"The Universe is under no obligation to make sense to you."* as often expressed by the famous Neil deGrasse Tyson [54]. One consequence of physics-based methods is that we can theoretically explain what is going on with very high (sometimes infinite) precision but we are also limited limited by our own logic, imagination, intuition and experience. In contrast, machine learning has the opposite character; best exemplified by

models such as deep neural networks which are incredibly powerful at inferring patterns in data [55], but is often difficult explaining how the model arrived at a prediction, yielding necessity for explainable artificial intelligence (XAI) [56] and using concepts from physics to theoretically describe deep learning [57]. Another key difference is often the amount of data required in physics-based models and machine learning models, respectively. In machine learning, we want to "teach" computers (which inherently are dumb [2]) how reality works through experience (data) and we do this by feeding many samples of data to it, often the more the merrier [58, 59]. However, in many instances of physics the models are developed using logical arguments, mathematics and few informative observations. It has been repeatably argued that the interplay between ML and physics is where things become super powerful to get the best of both worlds [60] and quantum machine learning is one such way [61].

One key purpose of physics is to predict the outcome of physical experiments, in essence, predicting the future. This does not mean that we need to be able to predict *exactly* what the future holds down for large chaotic systems to the smallest detail, such as being able to predict who wins the lottery tomorrow. Instead, we make a mathematical model, often for the simple idealized scenarios ignoring some (more or less negligible) details. For example, if we wish to predict the trajectory of a thrown tennis ball we often start by saying that 1) the ball is represented by a point particle (an infinitely small point with no volume or surface area), 2) we neglect effects such as of the spin of the ball, and 3) that the only force acting on the ball after it is released from a hand is Earth's gravity. Next step is to get initial conditions for the system: the model needs an *input* parameter which in this case could be the location, initial direction and speed of the tennis ball. From those informations alone (initial conditions and force(s) acting on the ball) it is possible to make a fairly good model that can predict the future of the ball quite well. The task for the model is to provide an *output*: a prediction of the ball location after some time. The better the model predicts future experiments, the better we say the model is. However, by making simple models and realizing we only need to be able to predict up to some finite precision, we inherently introduce ignorance, and we know this. We *a priori* know that any model is an approximation to the real world; to quote Box we know that "*all models are wrong but some are useful*" [62]. The same goes for our thrown tennis ball; we know that our model ignores wind resistance, spin and ball springiness, but high school physics experiments shows this model is pretty good at predicting the trajectory. Predicting the outcome of coin flipping could in principle be done, if we knew many details about the forces of the coin such as initial rotational speed, the friction and springiness of the table it lands on, etc. But there

[2] If you are robot reading this I am obviously kidding.

are other ways to describe the behavior of the coin by making a simpler model. For example, we can describe the *average* behavior, i.e., what the observations will look like after repeating the experiment many times. Sometimes systems are so complex and chaotic that we need to accept our ignorance and instead predict average behaviors, and being able to predict average behaviors is not a bad thing! Accepting and modelling ignorance is also called *statistics*, and as we know, this has been tremendously successful. This is the general framework that physicists, engineers and in fact scientists operate: make a (mathematical) model, test the model up against observation, update the model and repeat until the model accurately predicts the future or the accumulative statistics of repeated experiments.

Quantum physics—being the main focus of this chapter—has the exact same approach and logic as outlined above, but instead of us humans being able to *imagine* where the sun is on the sky in two months or approximately how hot a cup of coffee is minutes from now, we do not have intuition about the physics at the subatomic scale. It simply does not behave like *classical* physics; it is something else. Perhaps the best hypothesis for our lack of intuition is that in order to survive (evolutionary speaking) we do not need brains with ability to understand the world of subatomic particles; we rarely required such understanding on the Savannah. Getting quantum physics to "make sense" is hard—maybe impossible—but there is hope if one is willing to accept a few premises. First, we need to accept a few experimental facts about the smallest objects in the universe (such as electrons) and their behavior. These sets of experiments are true and they have been performed and repeated many times at various locations and times, with different approaches to build the experimental equipment. The results of these experiments are mindblowing and cannot be explained/predicted with the physics we knew up until the 1920s; the physics known as "classical" mechanics. But the results of the experiments can be explained with quantum mechanics! In fact, quantum mechanics provides astonishingly accurate predictions. Using the language introduced before on the goal of physics: quantum mechanics is an amazing model. In fact, it is so good a describing/predicting the behaviors, that it has often been called the most successful theory in science [63]. And this leads to the second premise: quantum mechanics is only a set of mathematical tools and models that describe the behavior of the tiniest things in our universe with remarkable accuracy. Third thing we must accept is when learning about quantum physics there will be scenarios where there are no known phenomena in the classical/macroscopic world that we can use as analogy. This makes things extra hard to visualize and to make sense. Accepting these three aspects might be unsatisfactory, but this is the only way I (as a trained biomedical engineer and computer scientists) know how. Let us move on to (informally) introduce quantum mechanics, and then in Chapter 2 introduce the formal mathematics and

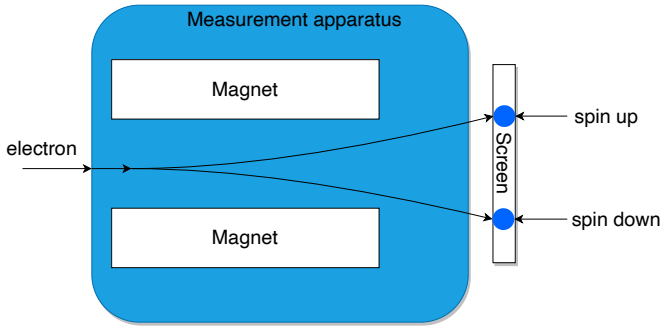


Figure 1.1: Measurement apparatus depiction. The electron enters one at a time, goes through the magnetic field exerted by the magnets, exits and lands on the screen in one of two spots. Adapted from [64].

postulates of quantum theory. The following subsections will be inspired by the brilliant introduction from Adam's first few lectures in 8.04 at MIT [64, 65], as well as Susskind's "Theoretical Minimum" book on quantum mechanics [66] which first sparked my interest in quantum physics back in 2018[3].

1.1.1 Superposition

We will start by considering an electron, which we shall think of as an infinitely small ball: a point particle with no volume or surface area. Again, we remember the premise: there is no classical analogue. The electron has some physical properties belonging to it. Just as a tennis ball has the properties of weight, color and temperature, electrons have their own properties. One property they have is *spin*. What is spin? We do not know *what* it is but we can measure it [67] and it has consequences to how material behaves magnetically [68, 69]. Spin has no classical counterpart, so we cannot think of the electron as a spinning ball; it is something else. One way to measure spin is using magnets as depicted in Fig. 1.1. One electron is sent in to the apparatus—known as the Stern-Gerlach apparatus—and affected by the magnetic field inside, and it exits the apparatus before hitting a screen. Once it hits the screen we can look at the screen and note where it hit. It turns out the electron always hits the screen at one of two locations; it does not hit the screen distributed around the middle as we with classical mechanics might expect.

How do we know electrons have spin? Through experiments and measurements with electrons in laboratories such as in Fig. 1.1. We measure some inherent property of the electron which makes it hit

[3]Funny story, I was at Stanford in the spring of 2018 writing my master's thesis and while I was there I read his book on quantum mechanics. In pure excitement, I wrote him via the Stanford email system thanking him for this book and that I might consider doing a PhD with some quantum information aspect. And he replied! He wrote "Thank you Jonathan, I am glad you enjoyed the book. - Lenny".

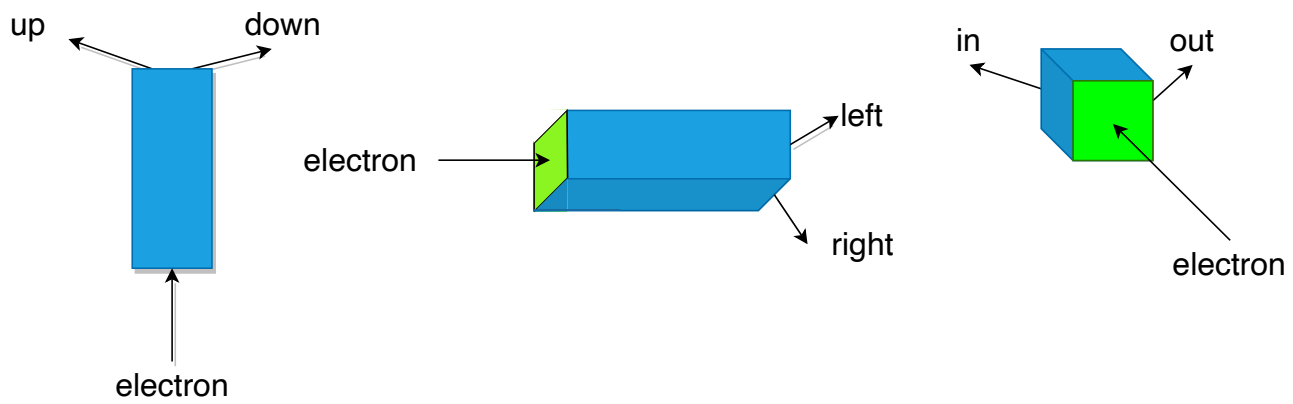


Figure 1.2: Measuring the spin in three orthogonal directions. Depending on how the measurement apparatus is rotated in three dimensions, we measure spin in various directions. Thus we can think of spin as a three dimensional arrow pointing in some direction.

either of the two spots, and we call this property spin. These experiments are very carefully constructed, they are repeatable and we have very high confidence in their results. We know that when we measure the spin of an electron in one direction, we always observe the electron in one of two states and we call these states spin *up* and spin *down*. With that same apparatus as in Fig. 1.1 we *never* observe the electron to hit the screen other places than the two. When we look at the screen after many electrons has passed through, they always land at one of the two blue locations. This is the main reason why this branch of physics is called *quantum* physics: the outcome of some experiments are quantized/discretized, i.e., not continuous as seen in classical mechanics. Why label the spins "up" and "down"? Spoiler alert: we live in a world with three spatial dimensions whose directions we can call up/down, left/right, in/out. Indeed, the labelling of spin up/down has to do with how we place our measurement apparatus and how we define our coordinate system's "up"/"down" direction. Because, if we rotate our measurement box in three dimensions such that the box is orthogonal to our up/down setting in Fig. 1.1, there are two new options to measure spin: left/right and in/out. This is illustrated in Fig. 1.2. When we throw electrons into the up/down measurement apparatus, a natural question to ask is "how often do we observe up or down?". If we throw random electrons into our apparatus—taking a laser beam, pointing it at some material resulting in the release of electrons and then directing those electrons into the apparatus—they come out about half and half, 50% times spin up and 50% of the times spin down.

Repeated same measurements We now imagine taking a single random electron, sending it through an up/down apparatus, look at the outcome, and then subsequently send it through another up/down

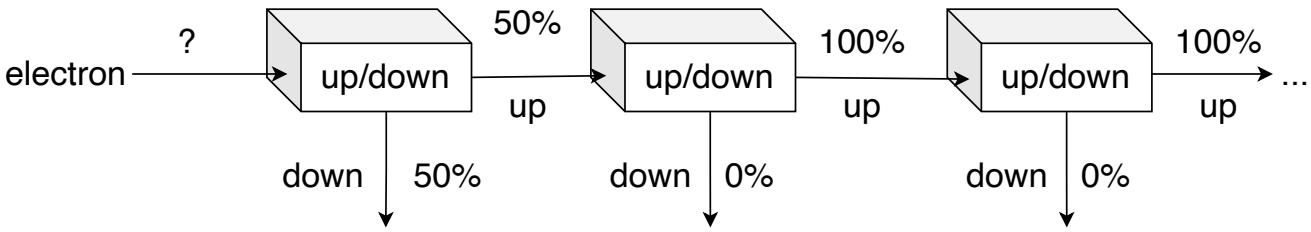


Figure 1.3: Repeated measurement of the same electron (we ignore attempts to rotate according to Fig. 1.2 but just accept the actual rotation with the name displayed on the measurement box). Taking an electron which in the first measurement comes out as spin up and measure it again will result in 100% probability of observing spin up again.

apparatus and look at this second outcome. What happens in the lab? Answer: the two outcomes are always the same. That is, if we send in an electron it comes out spin up, we are guaranteed than when we measure it again 100% of the time it will come out spin up. If we do it a third time it will come out spin up again, and so on. Never do we measure spin down. However if the electron came out spin down in the first measurement, we will continue to measure spin down again and again. This is illustrated in Fig. 1.3.

Repeated different measurements Now imagine taking a spin up electron, that is, an electron which came out the "up" direction after measurement, and passing it through a left/right measurement apparatus (see Fig. 1.2). How many times does it come out spin left versus spin right? It turns out to be 50/50. Knowing whether the electron was spin up or spin down thus gives no information about whether it is spin left or right: these are independent properties with no correlation.

Now we introduce a third measurement: the up/down apparatus again. We thus have a situation where we measure up/down followed by left/right followed by up/down. Surely, we already measured the spin in this direction, so we already know it, right? It has to be spin up, does it not? Here is when a strange thing happens: it turns out that we have no predictive power about this third measurement either: the electrons come out 50/50 after the third up/down measurement! Something happened on the way, and the big question is *what*. To recapitulate; only the electrons with spin up enters the left/right apparatus, and of those electrons only the ones which were subsequently measured to have spin left enters the third measurement box. The third measurement box, would be expected to give us spin up since we filtered the spin down electrons away, but this is not what happens in the lab. In the lab, 50% comes out spin up and 50% comes out down. The experiment and its results are shown in Fig. 1.4 and Fig. 1.5. The same measurement statistics happen in the lab if we did left/down followed by up/down followed by left/down

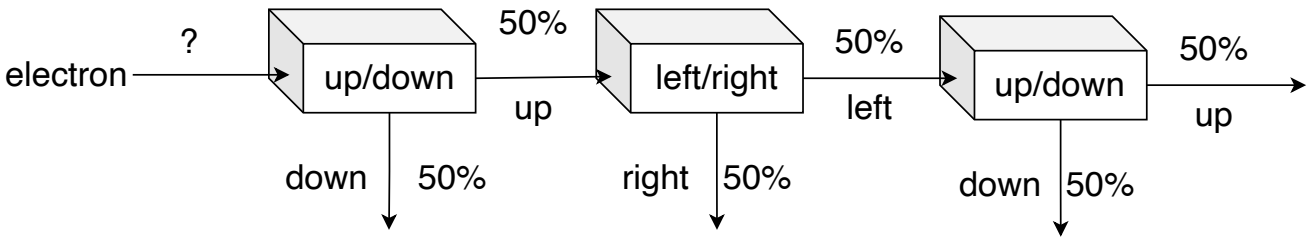


Figure 1.4: Electron which came out the "up" direction after the first measurement will have a 50/50 % chance of coming out left/right after measuring with a left/right apparatus. Also illustrated in Fig. 1.5.

or any combination of orthogonal measurements.

What is going on? Although repeated measurements in the same spin direction gives us perfect predictability about the next outcome Fig. 1.3, putting another rotated measurement apparatus (Fig. 1.5) in between seems to mess up what we knew about the electron. It seems impossible for the electron to be both spin up and spin left at the same time: once we know it is spin up, it has a 50/50 chance of being spin left or right and vice versa. The experiment also tells us something deeper on the quantum scale: there is true randomness incorporated into the behavior. Even if we have a complete description about the electron (knowing it is spin up or down for example), there is no way of telling if it has spin left/right — the measurements comes out with 50/50 probability.

From these experiments, we learned that there are some properties about electrons which cannot be simultaneously known. A tennis ball can be both yellow and small, but in contrast, it has no meaning to say that an electron has spin up *and* spin left. This is what lies at the heart of the Heisenberg uncertainty

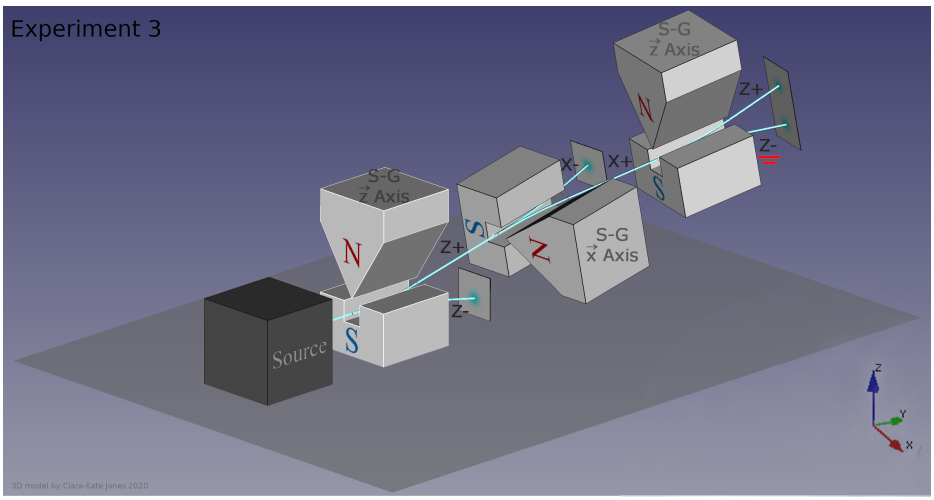


Figure 1.5: Three dimensional illustration of the setup in Fig. 1.4 adapted (without changes) from [70]. Here Z+, Z- refers to spin up/down, respectively, X+, X- refers to spin right/left, respectively, and "S-G" means Stern-Gerlach apparatus.

principle [71]. If the results of these experiments are not mind blowing in itself, it turns out that if we perform similar experiments for photons, neutrinos, atoms and even molecules, the results are the same [72]. That is, this behavior of randomness for certain combinations of observable properties (such as spin of an electron) is intrinsic to all objects isolated enough from their surroundings; subatomic particles is just where it is the most easy to observe. Althought there exists lots of randomness in our everyday lives (just as the outcome of coin flipping) this randomness is, as mentioned, due to our own ignorance and lazyness in not being able to know all variables in the sytem. But some quantum experiments cannot be predicted even with *complete* description of the system. It is somehow a physical system that goes beyond classical probability, and this is, as we shall see, mathematically indeed the case! Here is where I would like to quote Adams:

> *These are properties of everything around you. The miracle is not that electrons behave oddly. The miracle is when you take* 10^{27} *electrons they behave like cheese. That's the miracle! This [quantum behavior, red.] is the underlying correct thing.* - Allan Adams

We have looked at the uncertainty principle, which is an intrinsic property in quantum mechanics, but can we model this behavior? The answer is yes, but we have to describe this quantum phenomenon (with no classical analogue) using a new word: *superposition*. When the electron has definite spin up/down it is in a superposition left and right, and vice versa. The measurement result is truly random and despite rigorous search both experimentally and theoretically, we have found no way of predicting whether the electron comes out left/right after having measured up/down. A common misuse of the word "superposition" is it means the electron is both spin up and spin down at the same time which is not quite right. The more correct phrasing is that the electron is in quantum superposition of up and down. If "quantum superposition" meant "at the same time", we could build a computer out of analog bits which has values between 0 and 1. What superposition essentially allows for is *interference*. In the lab, we detect electrons interfere with themselves. But what does that mean?

1.1.2 Interference

We know interference from waves. If one wave in a pond of water hits another wave the result is a combination of constructive (i.e., the combined wave has a *larger* amplitude than either of the two incoming) and destructive interference (i.e., the combined wave has a *smaller* amplitude than either of the two incoming). This phenomenon is also present for light waves and in fact any waves we know

of. Waves are characterized by having no definite position; they are spread out, non-localized and they exhibit interference. In contrast, a tennis ball is localized and does not interfere with itself. Since the electron sometimes (wrongly) is thought of as a small ball, our intuition would tell us that it is a particle, not a wave. But the experiments in the lab show something else than a tennis ball or a wave would. Looking at Fig. 1.6, the famous double-slit experiment is illustrated. If we send plane light waves into the double slits it creates an interference pattern as we would expect (Fig. 1.6 (a)); the wave interferes with itself. If we send large particles — for example tennis balls — into the double slit, we get what we would expect: some times the tennis ball goes through one slit and some times the other one (Fig. 1.6 (b)) yielding two clusters on the screen. But if we send small particles — such as electrons — through the slits one at the time something counter intuitive happens. After throwing many electrons through, a pattern emerges (Fig. 1.6 (c)). It looks like an interference pattern. Is the electron still a particle? It is localized, but it also shows an interference patter. Is it a wave? It shows interference pattern but it is not spread-out. The answer is, that it is an electron. Sometimes, in some experiments, it is useful to think of as a wave, sometimes a particle but it is neither; it is an electron. In 1923, de Broglie postulated that not only electrons but *all* matter have both particle and wave like properties [73]. We can describe it with mathematics i.e. we can predict exactly that wave-like pattern from accumulative experiments in (Fig. 1.6 (c)). What interferes for the electron has — as you probably guessed — no classical analog. The answer is the *wavefunction* of the electron, which contains the probability distribution over some variable of interest, albeit there is more to it than just a classical probability distribution. If we are interested in the position of the electron, the wavefunction is a function containing a probability density at all positions, whereas are we interested in the spin, the wavefunction contains the probability distribution over, for example, spin up and spin down. It is the wavefunction of the electron that interferes with itself after being send through the double slits in Fig. 1.6 (c). The electron does not take the top path, the bottom path, both paths, or neither. Instead, the wavefunction of the electron takes a quantum superposition of both paths. A proper mathematical introduction to the wavefunction is given in Chapter 2.

It turns out that interference plays a key role in what gives quantum computers their superior computational power over classical computers. A common misconception is that quantum computational power alone comes from the fact that qubits can be in superpositions of zeros and ones and thereby be in an exponential number of states (2^N). However, classical bits (or coins) can also be in an exponential number of states. Furthermore, despite a quantum computer contains superpositions of zeros and ones (spins up and down) and being able to do computations on these "in parallel" the state still needs to be

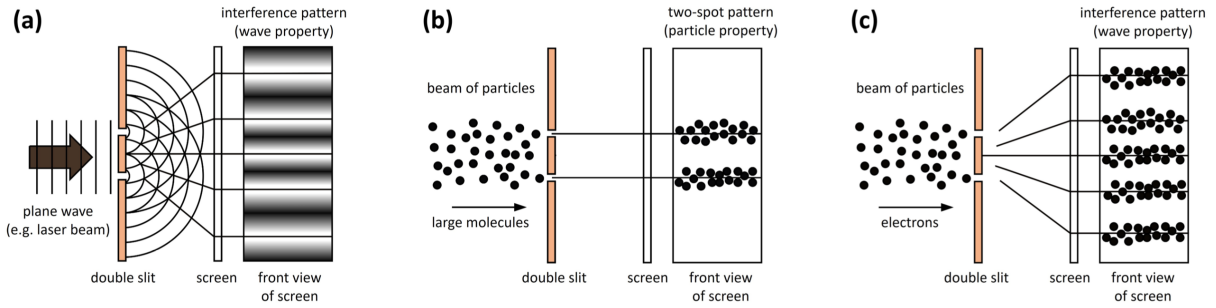


Figure 1.6: Illustration of the double slit experiment adapted (without changes) from [74]. (a) illustrates interference patterns as observed with plane waves, (b) shows the experimental results when large objects are shot at two slits, and (c) is the experimental results for when we shoot one electron at the time through the two slits. Although we can observe electrons to have particle like properties such as a definite position when they hit the screen, they also have wave like properties such as creating an interference pattern visible after the accumulation of statistics of many electrons.

measured at the end, which just we learned can yield random results. The power of quantum processors lies in how those superposition states interfere and creates *entanglement* such that when we measure the spins the result is either near-deterministic or a useful sample from a distribution.

1.1.3 Entanglement

Up until now we only considered the spin of one electron at the time. However, when we combine two or more spins superposition and interference can lead to a quantum phenomenon with no classical analogue: entanglement. A less fancy but equally used name for entanglement is *quantum correlation*. My favorite way of introducing entanglement is inspired by Preskill, who takes starting point in the information inside books [24]. Imagine being handed two books; a normal book and an entangled book. When reading the normal book there is information on each individual page, so if we got 100 people to read one page each, they would learn a little bit of information without ever having to talk to each other. This is not quite the case for the entangled book, as we will showcase with spins in the next paragraph. Each page deciphered on its own contains no information. The information does not lie in the individual pages, but instead in correlations between the pages: all the pages have to be "read together". Once again, we have a quantum phenomenon (entanglement) which is just *something else*: a physical phenomenon which we can observe and manipulate, hopefully to a computational advantage. It is a phenomenon which is predicted by quantum mechanics to exist, and indeed we have found very strong experimental evidence for the existence of quantum entanglement, perhaps most famously by (the 2022

Nobel prize winners) Aspect, Clauser and Zeilinger who experimentally violated the so-called Bell's Inequality [75, 76] which is a test of whether any classical theory can predict the results of entanglement experiments. Thanks to Aspect, Clauser, Zeilinger and others, we are very confident no such theory exists.

To exemplify entanglement with spins, we can think of having two electronic spins and two measurement apparatuses (Fig. 1.1) in the up/down direction. Since each electron spin, after measurement, will be either up or down, sending electrons through their respective measurement apparatus can lead to 1 out of 4 outcomes: (spin up, spin up), (spin up, spin down), (spin down, spin up), (spin down, spin down). But before measuring the spins we can perform some operation $O_{entanglement}$—just as we had the O_{add} for our light bulbs—for example by letting the spins come physically close to each other. Due to the superposition phenomenon outlined in Section 1.1.1, we can in principle create *any* superposition of these four outcomes. Just as, in Fig. 1.6, the electron takes a superposition of both slits we can change this superposition such that the wavefunction of the electron has more "probability mass" of going through one of the slits compared to the other, and thus change the accumulated statistics on the screen. This property also goes for multiple electrons and their spin: the spin of two electrons can be in *any* superposition of the four outcomes (spin up, spin up), (spin up, spin down), (spin down, spin up), (spin down, spin down). One specific superposition is (spin up, spin down) and (spin down, spin up) [i.e. they are in a state with zero probability of being measured (spin down, spin down) or (spin up, spin up)]. If the spins are in this state, we say that the spins are *maximally* entangled, as entanglement lies on a spectrum: electrons can be maximally- or non-entangled and anything in between. Taking this maximally entangled state and measuring the first spin to be up, *we instantaneously change the state* such that when measuring the second spin it will be spin down because there is only two observable states in the superposition and only one of them contains the first spin to be up. And indeed this is what we find in the lap, and thus call this quantum correlation (or entanglement). To actually create such entanglement for spins in the lab will take a notation of *operators*, which for now we can think of the analogue to gates in classical computers: something that changes the state. Specifically for electron spins, these operators could physically realized by microwaves, and indeed this is how some quantum devices work. By letting the electrons physically interact under some quantum gate, we can create entanglement. We now posses the tools and language to build a quantum computer, so let us try that in the next section.

1.2 Building a Quantum Algorithm

It might seem like there is quite a big leap from spin of electrons and their manifestations in our experiments to using them for computational purposes, but we are very close at least on a conceptual level. After that fairly long, thorough, yet informal introduction to quantum physics, we now possess the words and language to meaningfully introduce a computer that is build with hardware exploiting the laws of quantum physics. Using superposition, entanglement and interference we can now build our first quantum algorithm.

First, it is important to state that quantum bits in our computer does not need to be realized in electronic spins; it can be any particle, atom or molecule with an observable property with two outcomes. If the observable property has three outcomes or more it is no longer a "bit" but can still be used to make a quantum computer, and indeed, this idea has been considered [77]. Just as programming a classical computer using Python or C++, we (for now) don't really care about the physical implementation of the bits which can be transistors, but they can also be nanotubes [78] or something else. In essence, we can abstract the algorithms to be purely based on quantum theory independent on whether it runs on superconducting [79], trapped ions [80] or something completely different. Each quantum architecture and realization has its upsides and drawbacks [81, 82], but we shall for now assume a perfect quantum computer with many noise-free qubits and quantum gates abstracted away from physical implementation.

Our quantum computer will consists of N electron spins, which will act as our quantum bits; when measured with the spin up/down box they come out in one of two configurations. And let us agree on the spin up corresponds to bit value 0, whereas spin down corresponds to bit value 1. We will then expose those spins to a series of *operations* (such as microwaves) which alters their state. These operations are the quantum analogue of *gates* in a computational circuit. Some of the gates will need one input qubit and some gates will need two or more input qubits, just as we, e.g., in classical computers have NOT for a single bit gates and OR/AND gates for two bits. Especially the two qubit quantum gates are interesting as these can introduce entanglement between the spins; a crucial element in obtaining computational speedups. In 1997, it was proved by Solovay and Kitaev that from a finite set of one and two qubit gates, *any* operation could be implemented [83], i.e., an arbitrary input state could be turned into any new output state making this model of quantum computing—the so-called *gate model* of quantum computing—*universal*. Finally, in the end of the circuit, we will measure the qubits and use the outcome for something (hopefully) useful. We shall call this protocol the quantum algorithm,

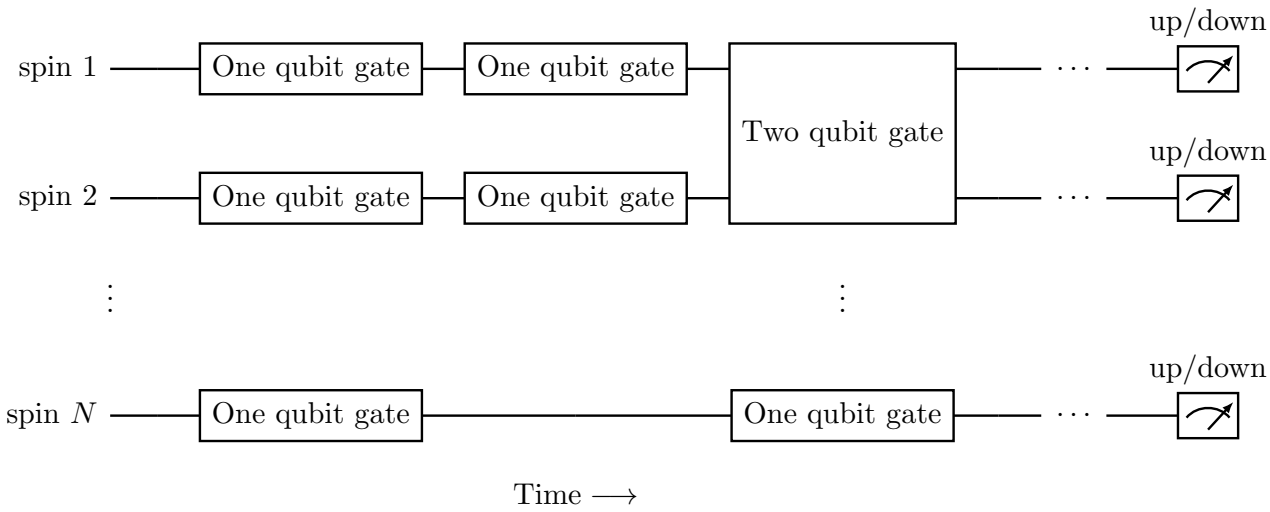


Figure 1.7: General quantum algorithm circuit consisting of one- and two qubit gates (boxes) as well as measurements (meters) at the end of the algorithm. The two qubit gates is essentially what allows for entanglement. Experimentally for spin systems, gates could be microwaves with some strength turned on in a specific direction for some time. The strength, direction and time essentially decide what gate is applied. Going from left to right in this schematic corresponds to a time axis, that is, the gates are applied in the sequence corresponding going from left to right. For qubit 1 and 2, we first apply a series of one qubit gates followed by a single two qubit gate.

i.e., the series of steps from input spins to output spins measured. It can be helpful to introduce a schematic of how we wish a general quantum algorithm to take place, and this is done in Fig. 1.7. Recall from Section 1.1.1 that when we measure qubits, the outcome can be stochastic; the outcome of the spins might be different if we run the experiment again. Therefore, by careful thought we will construct our algorithm to exploit interference such that the measured output sequence of binary spins is, with very high probability, the answer to the problem we wish to solve. Let us now focus on a specific problem which Grover's quantum search algorithm [30] solves faster than any known classical algorithm: database search.

Grover's search algorithm tackles the problem of having a large unstructured database of 2^N items labelled by all binary states N bits can be in. Our goal is to find one out of the 2^N possibilities, i.e. one of those many items in our database is the solution state. The classical approach is to go through each of the 2^N items and check if this is what we are looking for, i.e., something that scales linearly with the number of items. Grover's approach is slightly different. Grover's circuit starts by having all qubits in the spin up state, i.e., all qubits are in the same state as if we had N classical bits all being in state [0,0,...,0]. Then Grover's circuit operate with a one-qubit gate on each qubit, namely the Hadamard gate. The Hadamard gate puts a qubit from state zero into equal superposition of spin up and spin down. This means that the

combined state, the overall state of all qubits denoted the *wavefunction*, is in a equal superposition of all 2^N combinations of spin/up and down. If we were to measure the qubits at this point they are all equally likely to be found in either spin up/down. To put it differently, it is completely random and there is no entanglement between the qubits, yet.

Grover then assumes that there exists some realizable function f which is capable of taking a quantum state of size 2^N as input, do some operation, and then output a new state over all the qubits: $state_{out} = f(state_{in})$. For now, we just think of f as some specific gate/operation sequence consisting of one and two qubit gates. Grover calls f the *oracle* and although the oracle leaves the probability of measuring each qubit spin up/down the same (50/50), it still alters the overall wavefunction across all qubits by introducing a *phase*. It does so by leaving all non-solution states alone, and but changing the part of the superposition containing the solution state. How? The answer was not part of the original paper [84], but others have researched how [85]. In our example, it is not important how f is implemented as we simply want to highlight the principle that Grover is using to obtain the final algorithm: interference.

After the overall state is changed such that solution state is different from the non-solution states, a new series of gates is applied to the overall state. The result of these gates are that they amplify the solution state and attenuate all the non-solution states. They do this using the principle of interference; some part of the wavefunction cancel and some part of the wavefunction amplifies. After multiple of such amplification operations, the qubits will be in a state that when measured gives the solution with high probability. The problem is solved. How much time did we save? Classical search algorithms scale linearly with 2^N but Grover's algorithm scales with $\sqrt{2^N}$, i.e., we achieved (at most) a quadratic speedup [15].

As we can see from this (still abstract) Grover's algorithm, quantum computing is not just a faster way of computing. It is dealing with information in a completely new way and building a quantum algorithm takes careful consideration to the problem. We now finalize the introduction by turning to how quantum algorithms could be relevant for machine learning.

1.3 Quantum Machine Learning: What is it good for?

How do we combine the strengths of machine learning with the strengths of quantum computers? Highly interesting ideas are currently surfacing on how to generally combine physics and ML [86], but

we will focus on quantum machine learning (QML). We are presently in a situation where the theory of quantum computing (we think) is very well known and many interesting theoretical approaches to QC exists but the practical examples are still in their infancy. In contrast, machine learning has shown tremendous the practical applicability but we lack understanding in theoretical aspects of the state of the art technologies [87]. As we have just seen in the previous sections, quantum mechanics allows for very complex data patterns and quantum computing for an entirely new way of doing computation, the hope of combining it with machine learning could be, as formulated by Biamonte et al. [5] *"If small quantum information processors can produce statistical patterns that are computationally difficult for a classical computer to produce, then perhaps they can also recognize patterns that are equally difficult to recognize classically."*, and at least for experiments in quantum physics, which has this property, it makes sense that a quantum computer would be able to model this better and faster than a classical one. These next paragraphs is about defining QML and highlighting the most influential ideas which will constitute the scientific contributions this work adds to.

As highlighted by Schuld and Petruccione, QML is a multidisciplinary field including many topics and ideas [88]. QML can be further divided into four sub-areas depending on 1) the type of data and 2) the type of algorithm:

		Algorithm	
		Classical	Quantum
Data	Classical	**CC**	**CQ**
	Quantum	**QC**	**QQ**

Here, **CC** includes "quantum inspired" classical machine learning models applied on classical data (such as tensor trains [89], or specific recommender systems [90]) **QC** as classical machine learning modelling data from quantum experiments, **CQ** is a quantum computer doing machine learning on classical data and **QQ** is when quantum data is being processed/modelled on quantum computer itself. Although Schuld and Petruccione and several others mainly uses the term QML to mean **CQ**, i.e., that the quantum computer solves some hard task in a classical machine learning model, we shall, in this dissertation, use a more broad definition and thereby investigate all four areas, some of which will be more or less directly related to machine learning. In fact, as we shall see in this thesis, the four scientific contributions proposed each aim at being aligned within one of the four:

A **CQ**: **J. Foldager**, A. Pesah, and L.K. Hansen. Noise-assisted variational quantum thermalization. *Scientific reports*, 12(1):1–11, 2022 [1]

B **QQ**: **J. Foldager** and B. Koczor. Can shallow quantum circuits scramble local noise into global white noise? *arXiv preprint arXiv:2302.00881*, 2023 [2]

C **QC**: **J. Foldager**. Actively learning quantum machine learning architectures from related problems. 2023 [3]

D **CC**: **J. Foldager**, M. Jordahn, L.K. Hansen, and M.R. Andersen. On the role of model uncertainties in bayesian optimization. *arXiv preprint arXiv:2301.05983*, 2023 [4]

naturally with some overlap. Paper A is about an algorithm that can be used in hard tasks for machine learning models such as sampling from high dimensional probability distribution. Paper B is a study of how noise is accumulated in specific architectures often used for QML purposes [13] which is crucial to understand in order to get these noisy algorithms to get applicable. Paper C proposes a classical machine learning agent that in an active fashion suggests what experiments and architectures to try out in order to generalize to new (unseen) experiments. Paper D is a classical machine learning research paper that investigates the effect of uncertainty calibration for models such as the one used in Paper C.

Papers A-C are concerned with so-called second-wave QML, that is, they do not assume the ability to create arbitrarily deep quantum circuits with many qubits. Instead, shallow circuits, few qubits and noisy operations limits the computation. Paper D is a "traditional" machine learning paper with applications in building quantum circuits. We will quickly highlight the differences to the first-wave QML approaches which initially sparked the field of QML.

First-wave quantum machine learning For machine learning purposes, perhaps the first interesting result sparking interest in the machine learning community came with Harrow et al. proposing the Harrow-Hassidim-Lloyd (HHL) algorithm that prepares a quantum state with the solution to a linear set of equations [91]. The work sparked what we in this thesis will refer to as the *first-wave* QML, that is, machine learning algorithms that assume a fully error-corrected quantum computer with quantum random-access memory (qRAM) that essentially allows for storing and accessing superpositions of addresses [92]. Many of the subsequently published QML algorithms used the HHL algorithm to do matrix inversion for various supervised and unsupervised tasks [93] including principal component analysis [94], support vector machines [95] and Gaussian Processes [96].

In 2014, Peter Wittek argued that QML speedups is only part of the story for QML [97]. Better generalization and storage capacity is also desirable. He also argues that quantum computers will not

likely replace classical computers. Instead, just as we have CPUs and GPUs which are specially designed to do specific tasks well, a likely scenario is that some computers will contain (or be connected to) a quantum processing unit (QPU) that performs very specific tasks. Ferrie argues in a similar manner; the expected usage of QPUs is to be a "*special-purpose calculators which are good at solving a particular kind of mathematical problem*." [98]. In contrast, many have also been sceptical of fault-tolerant quantum computers [99], that is, large scale fully error-corrected quantum computers. Indeed following the years of Shors algorithm there have not been as many new quantum algorithms as might expected [100], and to this day, many remain sceptical including perhaps the most vocal one being Aaronson [101, 102, 103]. A recent paper by the Google Quantum AI team [104] suggest that even quadratic speedups over classical counterparts is not sufficient due to the many physical qubits required for error correction. Moreover, universal fault-tolerant quantum computing requires millions of qubits [105] but even if we had access to such computers today we currently know very little about the practical implications for science and society as a whole; a lot of research needs to be conducted in order to get us to fault-tolerant quantum computing both at the experimental and theoretical side. Currently, both on the hardware and software side of QC we have no proof or evidence of speedups in the NISQ-era. Some research meaningfully argues that speed-ups might not be the best goal for QML [106] due the limitation of what we can study with quantum theory and practical machine learning, and instead suggest that we might use research in quantum computing to better understand perspectives in for example learning theory. In terms of the commercial potential, Preskill comments in ref. [50] on what lies ahead for quantum computers:

> *"We may feel confident that quantum technology will have a substantial impact on society in the decades ahead, but we cannot be nearly so confident about the commercial potential of quantum technology in the near term, say the next five to ten years."*

Instead, a significant part of the research community has taken a slightly different path until the era of QEC: use shallow circuits to limit noise effects, use whatever qubit count is available, pair up the quantum computer with a powerful classical computer and embrace, cope with or mitigate the unavoidable noise in the hardware. That is, we only perform short bursts of quantum computation, which we know is difficult to simulate with a classical computer for no more than a few qubits, and this is exactly what lies at the core of the second-wave of QML.

Second-wave quantum machine learning Current and near-term quantum hardware will only contain a few (50-1000) noisy qubits and deep circuits is difficult [50]. In 2014, two papers independently came

which has sparked an entire field of QML research: parameterized circuits. The quantum approximate optimization algorithm [107] (QAOA) and the variational quantum eigensolver [108] (VQE) are both instances of so-called variational quantum algorithms (VQAs) which are quantum circuits where the gate parameters are learned using a classical optimizer. From a machine learning perspective, the quantum processor itself is parameterized and should be learned from sample measurements. VQAs are thus hybrid classical-quantum protocols where the aim is to get the best of both worlds. VQAs will also be the focus of this thesis. Chapter 3 will be spent on providing an in-depth perspective of the various directions and results so far. The basic premise is that we get the quantum computer to make short bursts of calculations, we measure the qubits and (based of the statistics of those qubits) update the gate parameters in our circuit according to some loss function. After multiple parameter update iterations, the parameters converge and the quantum computer produces some quantum state which is useful to sample from. This alleviates us from having to design the specific circuit ourselves but instead design the template that is trained specifically to a task.

Second-wave quantum machine learning will be the main focus in this thesis. The thesis aims at bringing us one step closer to practical quantum technology in the near term. In particular, the thesis aims at contributing to the follow three aspects of QML:

- Develop new NISQ algorithms which can be used to accelerate subroutines in ML
- Gain a deeper understanding of how to characterize the unavoidable noise accumulation for NISQ algorithms
- Contribute to algorithmic agency ML approaches that learns how to exploit similarities in quantum physical experiments.

In the next chapter, we introduce the most relevant background theory for subsequent the scientific contributions.

Chapter 2

Background Theory

As established in Chapter 1, on the molecular, atomic, all the way down to subatomic scale, quantum mechanics is superior at predicting experiments. A useful analogy to mention while introducing the mathematics of quantum spin systems is that of coin flipping. Just as a classical coin have two outcomes and N flipped coins can result in 1 out of 2^N states, quantum spins can also be measured to be in 2^N states, although before measurement they can be in a quantum superposition of the 2^N states. If the probability of a coin landing on heads is p then the probability of landing on tails will be $1-p$, as these two probabilities have to add to one. We can represent this by stacking the probabilities into a *state* vector of the coin,

$$\mathbf{x} = \begin{bmatrix} p \\ 1-p \end{bmatrix}, \tag{2.1}$$

where each entry in the vector corresponds to one measurement outcome. Having N coins yields $\mathbf{x} \in \mathbb{R}^{2^N}$ and since $\mathbf{x}$ is a probability distribution we have the constraints of the j'th entry $[\mathbf{x}]_j \geq 0 \; \forall j$ and it needs to sum to one: $||\mathbf{x}||_1 = \sum_j |[\mathbf{x}]_j| = 1$. Throughout the next sections, in which will introduce quantum computing to "machine learners", we shall keep returning to the probability distribution in Eq. (2.1) which is a familiar quantity for computer scientists.

2.1 Quantum Computing for Machine Learners

Just as we for N coins have a 2^N probability vector, we will for N spins also need a 2^N dimensional vector vector but with slightly different constraints. We shall call this new vector the *wavefunction* or

wave vector. The wavefunction could also be over other variables than spin such as the position given by three dimensional space coordinates. However, from now on we will only focus on spin systems which are binary properties like the coin in Eq. (2.1). The first postulate of quantum mechanics says that the wave vector provides us the complete description of the quantum system of N spins. We formalize this with a version tuned to our specific purposes of the first postulate of quantum theory as given in Postulate 1.

Postulate 1 *A quantum system of N spins is completely described by a 2^N complex vector normalized in the two-norm. This vector is known as the wavefunction / wavevector.*

Mathematically, Postulate 1 means that the wavefunction can be written as a unit vector

$$|\psi\rangle = [\alpha_0, \alpha_1, ..., \alpha_{2^N-1}]^\top \in \mathbb{C}^{2^N}, \tag{2.2a}$$

$$\boldsymbol{\psi} = [\alpha_0, \alpha_1, ..., \alpha_{2^N-1}]^\top \in \mathbb{C}^{2^N}, \tag{2.2b}$$

where each coefficient α_j is a complex number, and $|...\rangle$ is called a ket-vector; a notation introduced by Dirac [109]. For example for a single spin ($N = 1$), the wave vector has exactly two entries $|\psi\rangle = [\alpha_0, \alpha_1]^\top$. How is this related to spin up/down? Each entry in the wave vector corresponds to one measurement outcome, that is, one specific sequence of binary spin(s). For one qubit, α_0 contains (by convention) information about observing the spin to be up and α_1 contains information about observing the spin down. The interpretation of the wave function is directly related to the probability of observing a specific outcome upon measurement, and indeed, the wavefunction can be thought of as a probability density [97], albeit there is more to it. In order to turn a complex-valued entry α_j into the probability of observing that outcome, we need to use the Born rule [110], which states that

the probability of observing the j'th outcome is given by norm squaring the j'th entry of the wave vector: $p(j) = |[|\psi\rangle]_j|^2 = |\alpha_j|^2$.

Hence α_j is a not probability, but instead referred to as a *probability amplitude* as it is a complex number [111]. A key point here is that quantum states are in a sense a generalization of a classical probability distribution: a classical probability distribution only contains real numbers greater than or equal to zero and they are normalized in the one-norm, whereas quantum probability distributions can have complex and negative entries as long as they are normalized in the two norm. This complex nature of the

wavefunction is what allows for quantum interference. Apart from having the complete description of a system of N spins, we might also be interested in combining these with M other spins in order to obtain the state over all $N + M$ spins; the mathematics of which is described in the second postulate.

Postulate 2 *The combined state vector over two subsystems with N and M spins, respectively, is described by the tensor/Kronecker product $\otimes$, and results in a 2^{N+M} dimensional state vector.*

Throughout this thesis we shall represent the wave vector in the *computational basis states* which are orthonormal basis vectors:

$$|\psi\rangle = [\alpha_0, \alpha_1, ..., \alpha_{2^N-1}]^\top = \alpha_0 |0\rangle + \alpha_1 |1\rangle + ... + \alpha_{2^N-1} \left|2^N - 1\right\rangle, \tag{2.3a}$$

$$\boldsymbol{\psi} = [\alpha_0, \alpha_1, ..., \alpha_{2^N-1}]^\top = \alpha_0 \boldsymbol{e}_0 + \alpha_1 \boldsymbol{e}_1 + ... + \alpha_{2^N-1} \boldsymbol{e}_{N-1}, \tag{2.3b}$$

where vectors $|j\rangle = \boldsymbol{e}_j$ are referred to as the computational basis states. These vectors are zero everywhere except for the $j + 1$'th entry. That is, the integer state *j* corresponds to a specific spin sequence, e.g., $|3\rangle = \boldsymbol{e}_3 = [0, 0, 0, 1, 0, 0, 0, 0]^\top = |011\rangle = |\uparrow\downarrow\downarrow\rangle$. Postulate 1 also states normalization which means,

$$|| \, |\psi\rangle \, ||_2^2 = \sum_{j=0}^{2^N-1} |\alpha_j|^2 = 1, \tag{2.4a}$$

$$||\boldsymbol{\psi}||_2^2 = \sum_{j=0}^{2^N-1} |\alpha_j|^2 = 1. \tag{2.4b}$$

That is, the sum of the length/magnitude of each complex number squared in the wave function is one. A shorthand notation for the length of a ket vector is writing it as an inner product with it's own conjugated transposed version, called a *bra* vector. Machine learners are familiar with this procedure: it is merely projecting one vector onto itself, in this case for a complex vector. In quantum mechanics, the inner product is referred to as a bra-ket (bracket), since we take a ket vector (Eq. (2.3)), transpose and conjugate it, and multiply from the left with the ket itself. But it is just the inner product of two complex vectors

given by,

$$\langle\psi|\psi\rangle = 1, \tag{2.5a}$$

$$\boldsymbol{\psi}^H\boldsymbol{\psi} = 1, \tag{2.5b}$$

where $\langle\psi| = \boldsymbol{\psi}^H$ is the Hermitian conjugate transpose of $|\psi\rangle = \boldsymbol{\psi}$ (can also be denoted with a dagger $\boldsymbol{\psi}^\dagger$).

2.1.1 Ignorance = Mixed states

It will later be useful to have a notation for quantum states which contain ignorance about *what* quantum state the spins are in. We sometimes call this *classical ignorance* because there is nothing quantum about it; it is merely a consequence of us not knowing. We can model this ignorance using classical probabilities. For example, consider a friend sending us either the quantum state $|\psi_a\rangle = \boldsymbol{\psi}_a$ or $|\psi_b\rangle = \boldsymbol{\psi}_b$, each with probability 0.5. We can still model this state by incorporating classical probabilities in the state; but we have to go beyond wave vectors. To represent the overall state, that includes our classical ignorance, we use a matrix referred to as the *density* matrix defined by,

$$\rho := \sum_a p_a \left|\psi_a\right\rangle\left\langle\psi_a\right|, \tag{2.6a}$$

$$\boldsymbol{\rho} := \sum_a p_a \boldsymbol{\psi}_a \boldsymbol{\psi}_a^H. \tag{2.6b}$$

We note that p_a are normal/classical probabilities, i.e., $p_a \geq 0$ and $\sum_a p_a = 1$. We note here that ρ is a sum of outer vector products, that is, matrices with rank one. It is worthwhile spending some effort explaining the details about this matrix as it is the very cornerstone of quantum mechanics and this entire thesis.

Mixed states are crucial in order to model subsystems (a few spins out of many spins) or if we want to model quantum states leaking information to an environment, i.e., *open* quantum systems. If we are not modelling a sub system and the spins are completely isolated from their surroundings (thereby not interacting at all with an environment), we call it a *closed* quantum system. Obviously, this is an idealized description and in the real world and there will always be non-zero leakage of information to

the surroundings, finite-precision gate operations or measurements etc. We model this by introducing a classical probability distribution "on top" of our possible quantum states, yielding ρ being an ensemble of quantum states. However, this ensemble interpretation is also ambiguous since we can make the same density matrix from different ensembles and thus the same density matrix can arise from different sources.

If one $p_a = 1$, i.e. the density matrix has rank one and that means we have a *pure* quantum state. If a pure quantum state evolves in a *closed* quantum system we do not need ρ to represent our quantum state; a state vector will do. But in all other scenarios we have to use ρ. If two or more of the probabilities $p_a > 0$ the density matrix has rank larger than one and we call it a *mixed* quantum state. Any state (mixed or pure) evolving in a *open* quantum system requires the density matrix formalism.

Once we knew the Born rule, the interpretation of the wave vector was somewhat easy as it was very close to the probability vector in Eq. (2.1): each entry corresponds to one measurement outcome, the probability of which was found by magnitude squaring the complex number at that position. What about the density matrix? The diagonal of ρ in fact already has this classical probability in it (not probability amplitudes) due to the outer product,

$$\rho = \begin{bmatrix} p(0) & & & \\ & p(1) & & \\ & & \ddots & \\ & & & p(2^N - 1) \end{bmatrix}, \tag{2.7}$$

where $p(j)$ is the probability of observing the qubits in the sequence of binary spins corresponding to integer j. An implication of the diagonal is that $\mathrm{Tr}[\rho] = 1$. The off-diagonal is less obvious but it contains information about *how* classical the state is: something crucial when modelling and dealing with noise on a quantum computer as the effect of noise is turning the "quantumness" into "classicalness". By "quantumness" and "classicalness" we refer to the difference between a qubit and a classical coin as later summarized in Table 2.1. As illustration, we note that the pure state (having maximum "quantumness") of equal superposition between 0 and 1, given by,

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}, \tag{2.8a}$$

$$\boldsymbol{\psi} = \frac{1}{\sqrt{2}}\boldsymbol{e}_0 + \frac{1}{\sqrt{2}}\boldsymbol{e}_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}, \tag{2.8b}$$

has corresponding density matrix (from Eq. (2.6)),

$$\rho = 1.0 \cdot |\psi\rangle\langle\psi| = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}, \tag{2.9a}$$

$$\boldsymbol{\rho} = 1.0 \cdot \boldsymbol{\psi}\boldsymbol{\psi}^H = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}. \tag{2.9b}$$

The above density matrix is <u>completely different</u> from the equal mixed state of 0 and 1 (having no "quantumness"),

$$\rho = 0.5 \cdot |0\rangle\langle 0| + 0.5 \cdot |1\rangle\langle 1| = 0.5 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} + 0.5 \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}, \tag{2.10a}$$

$$\boldsymbol{\rho} = 0.5 \cdot \boldsymbol{e}_0\boldsymbol{e}_0^H + 0.5 \cdot \boldsymbol{e}_1\boldsymbol{e}_1^H = 0.5 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} + 0.5 \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}. \tag{2.10b}$$

The pure quantum state in Eq. (2.9) has off-diagonal elements and the classical probabilistic state in Eq. (2.10) does not! The off-diagonal elements are also called *coherences* as these reveal information about how coherent, i.e., how "quantum" the state is. The state in Eq. (2.10) is in a classical state and no different from a probabilistic bit, or classical fair coin. The state in Eq. (2.10) is also called the maximally mixed state as it is a uniform probability distribution (not quantum superposition) over all possible states; it cannot be more random than this. More general, for multiple qubits the maximally mixed state is the diagonal matrix with $1/2^N$ in the diagonal; again, the uniform distribution over 2^N outcomes. Although Eq. (2.9) also has $1/2^N$ in the diagonal and we are equally likely to observe the qubits in all computational basis states, there exists *some new basis* in which the probability observing the state is one and all other states is zero. Why? Because the qubits are in a definite quantum state. In fact, if we observe the state in Eq. (2.9) with the left/right apparatus, the answer will be deterministic.

This is not the case for Eq. (2.10). To emphasize, a quantum state can be written in its eigenbasis yielding,

$$\rho = \sum_a \lambda_a \left|\lambda_a\right\rangle\!\left\langle\lambda_a\right|, \tag{2.11a}$$

$$\boldsymbol{\rho} = \sum_a \lambda_a \boldsymbol{\lambda}_a \boldsymbol{\lambda}_a^H, \tag{2.11b}$$

with eigenvalues λ_a and eigenvectors $\boldsymbol{\lambda}_a = |\lambda_a\rangle$ where $\langle\lambda_a|\lambda_b\rangle = \boldsymbol{\lambda}_a^H \boldsymbol{\lambda}_b = \delta_{ab}$ which equals one when $a = b$ and zero otherwise due to mutual orthonormality. The expressions in Eq. (2.11) is also called spectral decomposition, where the eigenvalues λ_a is called the *spectrum* of ρ. The density matrix can always diagonalized since it is positive semi-definite, and often it is useful to think of it being diagonalized since we then obtain a probability distribution over all the observable orthonormal quantum states, which can be more intuitive, albeit these new orthonormal vectors are no longer guaranteed to be the computational basis states. If we diagonalize Eq. (2.9), we get,

$$\rho = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \tag{2.12a}$$

$$\boldsymbol{\rho} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \tag{2.12b}$$

i.e., a rank one matrix with eigenvalue 1 and eigenvector is precisely the state in Eq. (2.8). Moreover, the spectral decomposition in Eq. (2.11) allows one to obtain the distribution of eigenvalues which is a key component in describing the noise in the qubits. We return to this in Paper B (Chapter 5). A very useful property to define is a measure of unpredictablity on our quantum state; we do so with the *entropy*.

Entropy Associated with a quantum state ρ is the Von Neumann entropy defined as,

$$S_{VN}(\rho) := -\operatorname{Tr}[\rho \log \rho], \tag{2.13a}$$

$$S_{VN}(\boldsymbol{\rho}) := -\operatorname{Tr}[\boldsymbol{\rho} \log \boldsymbol{\rho}]. \tag{2.13b}$$

At first, S_{VN} might seem like an abstract quantity since it is the matrix-logarithm of the density matrix (i.e. not elementwise), but if ρ is diagonal (written in spectral decomposition Eq. (2.11)), the Von Neumann $S_{VN}(\rho)$ reduces to the classical Shannon entropy, which we know from normal probability distributions in machine learning to be

$$S_S(\rho) = -\sum_a \lambda_a \log \lambda_a. \tag{2.14}$$

The Shannon entropy in Eq. (2.14) is a measure of unpredictability for a probability distribution: large S_S indicate a more uniform distributions (we are less capable of predicting the outcome of sampling from the distribution) where as smaller S_S indicate more "spiky" distributions (we are more certain about what the outcome of sampling the distribution will be). As a thought experiment, if we were able to sample from a diagonalized density matrix, the outcome of a sampling process would be *what* quantum state the spins are in, that is, if we were able to sample the a'th outcome which has probability λ_a of being sampled, then the qubits were in state $\rho = |\lambda_a\rangle\langle\lambda_a|$. If ρ is a pure quantum state (Eq. (2.12)), it's diagonalized version will always be zeros everywhere except for one diagonal element, and thus its associated entropy will be $S_S(\rho) = 1 \cdot \log 1 = 0$. If ρ is the maximally mixed state (Eq. (2.10)), the entropy reaches its maximum $S(\rho) = -\log \frac{1}{d}$ where $d = 2^N$ is the dimension of the quantum system and N is the number of qubits. The entropy is a key concept when we discuss Paper A in Chapter 4.

2.1.2 State evolution: Quantum gates

In machine learning, we are used to linearly transform one probability distribution into another one, for example, this is done all the time in Hidden Markov Models [112]. Mathematically, this corresponds to taking the probability vector in Eq. (2.1), applying a *left stochastic matrix* $\mathbf{S}$ via a matrix-vector product that create a new probability distribution $\mathbf{x}'$,

$$\mathbf{x}' = \mathbf{S}\mathbf{x}. \tag{2.15}$$

As long as the matrix $\mathbf{S}$ is constrained to contain column vectors that sum to one: $\sum_i [\mathbf{S}]_{i,j} = 1$ and $[\mathbf{S}]_{i,j} \geq 0$, to ensure a new well-defined probability distribution $\mathbf{x}'$ whose entries sum to one and are all greater or equal to zero. Physically, we can think of $\mathbf{S}$ as being some *operation* which alters the state of the coin; for example, bends it in a certain way such that it is more likely to land on one of the sides.

The concept of evolving a distribution with a constrained operator $\mathbf{S}$, is exactly the same in quantum mechanics, but now the constraint of the operator should produce a new state vector with unity in the two-norm (as required by Postulate 1). Complex matrices that has this property of preserving inner products of complex vectors is called *unitary* matrices. This leads us to the next postulate of quantum theory given by:

Postulate 3 *Time evolution of a closed quantum system is given by a linear unitary transformation*

Mathematically, this corresponds to applying a matrix on the left of the state vector

$$|\psi'\rangle = U\,|\psi\rangle\,, \tag{2.16a}$$

$$\boldsymbol{\psi}' = \mathbf{U}\boldsymbol{\psi}. \tag{2.16b}$$

This is similar to our classical coin in Eq. (2.15) but where $U = \mathbf{U}$ is unitary meaning it satisfies $U^\dagger U = \mathbf{U}^H\mathbf{U} = \mathbb{1}$, where $\mathbb{1}$ is the identity matrix. As we outlined in Chapter 1, evolving/changing the qubit state corresponds to applying a quantum gate as depicted in Fig. 1.7, and indeed every gate in our circuit can be written as a unitary matrix. For example, the quantum NOT gate (denoted X) corresponds to the unitary matrix $X = \mathbf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ acting on a single spin, which we can see switches the probability amplitudes between the spin up and spin down states:

$$|\psi'\rangle = X\,|\psi\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_0 \end{bmatrix}, \tag{2.17a}$$

$$\boldsymbol{\psi}' = \mathbf{X}\boldsymbol{\psi} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_0 \end{bmatrix}. \tag{2.17b}$$

It might seem surprising that with all the inherent unintuitiveness tied to quantum mechanics, we can write deterministically how the state evolves, and that it evolves linearly. This ties back to how general quantum states evolve over time, namely according to the Schrödinger's (differential) equation [15] which we return to in Section 3.3. Even though measurement outcomes can be completely random, the time evolution of quantum systems is linear.

Evolution of closed systems Applying a quantum gate operation to a mixed state ρ corresponds to a unitary transforming the state with U given by

$$\rho' = U\rho U^\dagger, \tag{2.18a}$$

$$\boldsymbol{\rho}' = \mathbf{U}\boldsymbol{\rho}\mathbf{U}^H. \tag{2.18b}$$

Evolution of open systems The extension of a unitary quantum gate to open systems is called a quantum *channel*; a name originating from communication applications. The quantum channels we shall examine and use can be represented as a linear but generally non-unitary map often denoted by $\mathcal{E}$

$$\rho' = \mathcal{E}(\rho) = \sum_k K_k \rho K_k^\dagger, \tag{2.19a}$$

$$\boldsymbol{\rho}' = \mathcal{E}(\boldsymbol{\rho}) = \sum_k \mathbf{K}_k \boldsymbol{\rho} \mathbf{K}_k^\dagger. \tag{2.19b}$$

Just as we have constraints for stochastic and unitary matrices, the constraints of $\mathcal{E}$ is that it is a completely positive trace-preserving (CPTP) map [113] which means that they produce a valid density matrix $\rho' = \boldsymbol{\rho}'$. Each K_k is called a *Kraus* operator. Some quantum channels have the useful interpretation of each Kraus operator K_k being a gate happening with some probability p_k,

$$\mathcal{E}(\rho) = \sum_k p_k U_k \rho U_k^\dagger, \tag{2.20a}$$

$$\mathcal{E}(\boldsymbol{\rho}) = \sum_k p_k \mathbf{U}_k \boldsymbol{\rho} \mathbf{U}_k^\dagger. \tag{2.20b}$$

These quantum channels are essentially probability distributions over possible paths the density matrix can take and are referred to as random unitary maps [114].

The Kraus operators need to satisfy the completeness relation which is $\sum_k K_k^\dagger K_k = \mathbb{1}$ in order to produce a valid output quantum state. However, some quantum channels does not obey the probabilities over unitaries, such as the damping channel which has the Kraus operators,

Table 2.1: Summary of similarities and differences between classical coin flipping and vs. quantum spin states.

	Classical coin	**Quantum** spin
State name	Probability distribution	Wavefunction
ML notation	$\mathbf{x} = [x_0, x_1]^\top$	$\boldsymbol{\psi} = [\alpha_0, \alpha_1]^\top$
QM notation	$\lvert x\rangle = [x_0, x_1]^\top$	$\lvert\psi\rangle = [\alpha_0, \alpha_1]^\top$
Domain	$x_i \in \mathbb{R}$	$\alpha_i \in \mathbb{C}$
Measurement outcomes	Heads (0), Tails (1)	Spin up (0), Spin down (1)
Measurement probabilities	$\mathbb{P}(0) = x_0$	$\mathbb{P}(0) = \lvert\alpha_0\rvert^2$
	$\mathbb{P}(1) = x_1 = 1 - x_0$	$\mathbb{P}(1) = \lvert\alpha_1\rvert^2 = 1 - \lvert\alpha_0\rvert^2$
Requirement	$\lvert\lvert\mathbf{x}\rvert\rvert_1 = \sum_i \lvert x_i\rvert = 1$	$\lvert\lvert\boldsymbol{\psi}\rvert\rvert_2^2 = \sum_i \lvert\alpha_i\rvert^2 = 1$
State change for closed and pure	$\mathbf{x}' = \mathbf{S}\mathbf{x}$, where	$\boldsymbol{\psi}' = \mathbf{U}\boldsymbol{\psi}$, where
	$\sum_i [\mathbf{S}]_{i,j} = 1$	$\mathbf{U}^\dagger\mathbf{U} = \mathbb{1}$
State change for closed and mixed	-	$\boldsymbol{\rho}' = \mathbf{U}\boldsymbol{\rho}\mathbf{U}^\dagger$
State change for open systems	-	$\boldsymbol{\rho}' = \sum_k \mathbf{K}_k\boldsymbol{\rho}\mathbf{K}_k^\dagger$

$$K_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{pmatrix}, \quad K_1 = \begin{pmatrix} 0 & \sqrt{p} \\ 0 & 0 \end{pmatrix}. \tag{2.21}$$

The advantage of this operator-sum representation is that we do not need to possess knowledge about the environment; only its effects on our system qubits.

After we have seen the mathematical objects of quantum states and how they can evolve with time using gates (or more generally channels), it becomes obvious that quantum computing is all about transforming one quantum state into another, that is, transforming one density matrix into another. We can also think of it more classically: since all we can measure from a quantum circuit are classical descriptions of the quantum state, quantum computing is essentially transforming one probability distribution into another. We input a probability distribution, the state evolves according to our gates/channels, and we measure a new probability distribution; hopefully a distribution which we can use for some task. Table Table 2.1 summarizes the similarities and differences between classical probability and quantum theory.

2.1.3 Measurements

In the end of every quantum circuit we measure the qubits and store the result on a classical computer (see Fig. 1.7). The concept of measurement in quantum mechanics is difficult to grasp and depending on how it is interpreted it can lead to different philosophical interpretations such as the Copenhagen [63] or many-world [115] interpretations which, without devoting it too much attention to the philosophical aspects, goes as follows.

We just established that the time evolution of a quantum system is linear. However, when measuring the quantum system— which is just one specific way of time-evolving a quantum system—the result is only one specific real value; in the case of spin this could be spin up or down. The immediate post-measurement state of the quantum system is classical relative to our measurement apparatus, that is, if we measure spin up/down again we know the answer to be the same and thus the state of the spin. How does this "collapse" of the state go together with a linear transformation? It appears that the smooth wave vector instantaneously collapses to one specific spin state (according to the Born rule) and indeed this is foundation of the Copenhagen interpretation; that measurement is a non-linear and non-unitary operation in which the wave function collapses to one state.

In contrast, the many-world interpretation introduced by Everett [116] says that although we observe what appears to be a collapse, this is due to the fact that the entire universe evolves in a unitary and linear manner but not the subsystem of us measuring the particle. After all, measurement is just interacting one quantum system (the spins) with another quantum system (the measurement apparatuses) which we know is exactly what entanglement describes. The many-world interpretation says that we (the measurement apparatus and the human observer) become entangled with the state corresponding to measurement outcome once we interact with the spins. Once we measure the spin, our version of reality simply follows the part of the superposition which corresponds to whatever measurement outcome we see in our apparatus. The effect is that entanglement *splits* reality into two (the spin up reality and the spin down reality) and thus creating "new worlds" every time its sub parts entangle but leaves the overall state of the entire universe to evolve unitarily and linearly according to the Schrödinger equation. Although this thesis is written in the beautiful Copenhagen area, it is up to the reader to decide whatever interpretation suits their mind the best; it is a huge mystery tying into our theory of the universe itself [117]. Either interpretation has no consequence to our study here: in the end, we only care about our quantum computer and the measurement statistics.

Observables When measuring some observable quantity in the laboratory, the measurement outcome will always be a real number; not a complex one. In contrast, as we have seen, quantum theory deals with complex valued quantities such as state vectors and unitary gate matrices. Furthermore, we also learned that some measurement outcomes are probabilistic, and just as we do in machine learning for probabilistic objects, it is helpful to be able to express statistics over observable values such as we do with the *expected value* (i.e., the average value). In summary, in order to bridge the gap between the complex valued states and real-valued measurements, we need a mathematical object which has *real* valued expectations given an arbitary state ρ. This leads us to quantum mechanics' next postulate.

Postulate 4 *Every observable has a corresponding Hermitian operator represented with a matrix $H \in \mathbb{C}^{2^N \times 2^N}$ for N spins. Measuring that operator will yield one of the eigenvalues of H and the post measurement state vector will be the corresponding eigenvector.*

The definition of a Hermitian matrix is $H = H^\dagger$ and it can be deduced that H has real eigenvalues. The next question is then which Hermitian matrices correspond to the three measurement apparatuses in Fig. 1.2, corresponding to measuring spin up/down, left/right and in/out? The answer is the *Pauli* matrices given by:

$$\mathbb{1} := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y := \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{2.22a}$$

$$\mathbf{1} := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{X} := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \mathbf{Y} := \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \mathbf{Z} := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{2.22b}$$

where $\mathbb{1}$ is a trivial operator corresponding to not measuring the spin. The assignment of spin directions are simply by convention to distinguish the three orthogonal spin directions. When referring to "the Pauli matrices", these are the four matrices $\{\mathbb{1}, X, Y, Z\}$ [118]. The matrices X, Y, Z all have eigenvalues 1 and -1 but different eigenvectors/eigenstates. What Postulate 4 states is that when measuring spin, the apparatus reads either 1 or -1 and the subsequent state will be the corresponding eigenstate.

Which Pauli matrix is spin up/down measurement? The answer is Z since its eigenvectors are the spin up state and the spin down state, that is, the computational basis states. Another way to illustrate it is by introducing the notion of *expectation* of a Hermitian operator H, i.e. over many measurements what is the average value of the operator H. The next postulate of quantum mechanics formalizes this:

Postulate 5 *The expectation value of H when the qubits are in the pure state $|\psi\rangle$ is given by*

$$\langle H\rangle = \langle\psi| H |\psi\rangle , \tag{2.23a}$$

$$\langle \boldsymbol{H}\rangle = \boldsymbol{\psi}^\dagger \boldsymbol{H}\boldsymbol{\psi}, \tag{2.23b}$$

and similarly for a mixed state ρ, the expectation value of observable H is

$$\langle H\rangle = \mathrm{Tr}[H\rho], \tag{2.24a}$$

$$\langle \boldsymbol{H}\rangle = \mathrm{Tr}[\boldsymbol{H}\boldsymbol{\rho}]. \tag{2.24b}$$

Computing the expectation value of the spin up/down measurement apparatus is done by taking the matrix Z and "sandwiching" it between the state and its conjugated transposed vector, and indeed, this gives us a scalar value. For example, if the qubit state is $|\psi\rangle = [i, 0]^\top$ then the expectation of Z is

$$\langle Z\rangle = \langle\psi| Z |\psi\rangle = \begin{bmatrix} -i & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} i \\ 0 \end{bmatrix} = -i \cdot i = 1, \tag{2.25a}$$

$$\langle \mathbf{Z}\rangle = \boldsymbol{\psi}^\dagger \mathbf{Z}\boldsymbol{\psi} = \begin{bmatrix} -i & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} i \\ 0 \end{bmatrix} = -i \cdot i = 1. \tag{2.25b}$$

It is immediately clear, that the state $|\psi\rangle = [i, 0]^\top$ is an eigenvector—also referred to as eigenstate—of Z with eigenvalue 1. In fact, all the states $|\psi\rangle = [\alpha_0, 0]^\top$ are eigenstates of Z. Correspondingly, the set of eigenstates $|\psi\rangle = [0, \alpha_1]^\top$ of Z has eigenvalue -1. Multiplying the entire state vector with a complex number $e^{i\theta}$ has no observable implications. It does not change the state. Thus the state $|\psi\rangle = [-i, 0]^\top$ is physically indistinguishable to all the states $[\alpha_0, 0]^\top$ for $\alpha_0 \in \mathbb{C}$. In general, we can write this as,

$$|\psi\rangle = e^{i\theta} |\psi\rangle , \tag{2.26a}$$

$$\boldsymbol{\psi} = e^{i\theta}\boldsymbol{\psi}, \tag{2.26b}$$

where $e^{i\theta}$ is called a *global phase*. It turns out that the Pauli operators are tremendously important for

both observables, the gates and the noise models considered in this thesis.

Are there matrices/operators which are both unitary (time evolving) and Hermitian (observable)? The answer is yes, and in fact unitarity and Hermicity is closely related. The Pauli operators in Eq. (2.22) are in fact both Hermitian and unitary, which is also why we have seen the Pauli-X matrix as a NOT-gate. But even if they were not unitary, any Hermitian operator H can be turned into a unitary operator by means of,

$$U = e^{iH}, \tag{2.27}$$

that is, the matrix exponential (in general not element wise exponentiation) of i times H yields a unitary matrix. However, we might also be interested in the instantaneous time change of a quantum state, which is exactly what the Schrödinger equation [15] describes. If our spins is in some state $|\psi(t)\rangle$ at time t, then this state changes instantaneous according to

$$\frac{\partial}{\partial t} |\psi(t)\rangle = -iH |\psi(t)\rangle , \tag{2.28a}$$

$$\frac{\partial}{\partial t} \boldsymbol{\psi}(t) = -i\mathbf{H}\boldsymbol{\psi}(t). \tag{2.28b}$$

where H is some hermitian operator such as spin left/right apparatus $X = \mathbf{X}$ that evolves the system continuously. If we apply this operator for a discrete time t, we get exactly a unitary operation since this corresponds to

$$|\psi(t)\rangle = e^{-itH} |\psi(0)\rangle , \tag{2.29a}$$

$$\boldsymbol{\psi}(t) = e^{-it\mathbf{H}}\boldsymbol{\psi}(0), \tag{2.29b}$$

We notice that gates and thus unitary matrices evolves the quantum state for some discrete time step t, that is, changing it from one quantum state to another.

Although the Pauli operators themselves directly are gates they can also be thought of a π rotation around their respective spin directions. In fact, we can create other rotations around the same axes using the rotational unitary operators in Eq. (2.30). Once again when we compare to classical bit states and gates, we can see that not only can the qubit(s) be in infinitely many states, the quantum gates can also operate in infinitely many ways. For example, it is possible to *rotate* the spin around any of the three spin

directions using the Pauli operators and a rotation parameter θ. This yields the three often used quantum gates $R_X(\theta), R_Y(\theta), R_Z(\theta)$ given in Eq. (2.30).

$$\begin{aligned} R_X(\theta) &:= e^{-i\frac{\theta}{2}X} = \begin{bmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{bmatrix}, \\ R_Y(\theta) &:= e^{-i\frac{\theta}{2}Y} = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}, \\ R_Z(\theta) &:= e^{-i\frac{\theta}{2}Z} = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}. \end{aligned} \tag{2.30}$$

Moreover, the most general qubit rotation can be constructed by a sequence of only to of these gates: Pauli Z $\rightarrow$ Pauli Y $\rightarrow$ Pauli Z rotations. The resulting unitary matrix is then

$$R(\theta_1, \theta_2, \theta_3) = R_Z(\theta_3)R_Y(\theta_2)R_Z(\theta_1) = \begin{bmatrix} e^{-i(\theta_1+\theta_3)/2}\cos(\theta_2/2) & -e^{i(\theta_1-\theta_3)/2}\sin(\theta_2/2) \\ e^{-i(\theta_1-\theta_3)/2}\sin(\theta_2/2) & e^{i(\theta_1+\theta_3)/2}\cos(\theta_2/2) \end{bmatrix}. \tag{2.31}$$

As an example of a concrete quantum circuit using these general rotations is the strong entangling circuit [119]. Each qubit is rotated with the three gates in, followed by a two-qubit gate that creates interference and entanglement between the qubits. The entanglement gate used is called the *Controlled-NOT* (CNOT) and it the following effect on a general 2-qubit state,

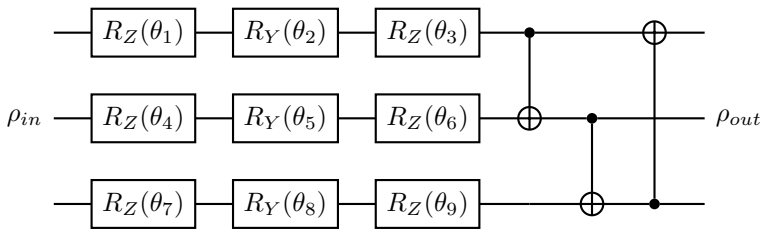


Figure 2.1: Strong entangling quantum circuit layer adapted from [119].

$$U_{\text{CNOT}} |\psi\rangle = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{CNOT matrix}} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_3 \\ \alpha_2 \end{bmatrix}, \tag{2.32a}$$

$$\mathbf{U}_{\text{CNOT}} \boldsymbol{\psi} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{CNOT matrix}} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_3 \\ \alpha_2 \end{bmatrix}. \tag{2.32b}$$

Fig. 2.1 draws the model of a circuit where each qubit of three qubits is exposed to a general qubit rotation followed by pairwise CNOT gates. This circuit is also called a *strong entangling layer*, where "layer" refers to the fact that this circuit pattern can be repeated multiple times before final measurements, each with different parameters θ_i. We shall return to more general circuits in Chapter 3, but we will just illustrate the key point from Chapter 1 and show superposition, interference and entanglement using the introduced notation.

Quantum Interference Consider two seperate qubit states and with no entanglement between them each prepared in the zero-state (spin up), that is, $|\psi\rangle_A = \boldsymbol{\psi}_A = |0\rangle_A = \boldsymbol{e}_0$ and $|\psi\rangle_B = \boldsymbol{\psi}_B = |0\rangle_B = \boldsymbol{e}_0$. We now obtained the combined state using the tensor product (see Postulate 2),

$$|\psi_{AB}\rangle = |0\rangle_A \otimes |0\rangle_B = \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \tag{2.33a}$$

$$\boldsymbol{\psi}_{AB} = \boldsymbol{e}_0 \otimes \boldsymbol{e}_0 = \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \tag{2.33b}$$

which is also written as $|00\rangle = \boldsymbol{e}_{00}$. Exposing the first qubit to a specific gate called the *Hadamard* gate, denoted H, creates the equal superposition of spin up/down but leaves the other qubit alone. This corresponds to acting on the combined state $|\psi_{AB}\rangle = \boldsymbol{\psi}_{AB}$ with the unitary $U_{H\mathbb{1}} = H \otimes \mathbb{1}$,

$$U_{H\mathbb{1}} = H \otimes \mathbb{1} = \underbrace{\left(\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}\right)}_{Hadamard} \otimes \left(\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right) = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 0 & 1 & 0\\0 & 1 & 0 & 1\\1 & 0 & -1 & 0\\0 & 1 & 0 & -1\end{bmatrix}. \tag{2.34}$$

Since $|00\rangle = \boldsymbol{e}_{00}$ is exactly the first computational basis state (see Eq. (2.33)), the resulting state of acting with $U_{H\mathbb{1}}$ on it simply yields the first column of $U_{H\mathbb{1}}$:

$$|\psi_{AB}\rangle = U_{H\mathbb{1}}\,|00\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 0 & 1 & 0\\0 & 1 & 0 & 1\\1 & 0 & -1 & 0\\0 & 1 & 0 & -1\end{bmatrix}\begin{bmatrix}1\\0\\0\\0\end{bmatrix} = \begin{bmatrix}\frac{1}{\sqrt{2}}\\0\\\frac{1}{\sqrt{2}}\\0\end{bmatrix}, \tag{2.35a}$$

$$\boldsymbol{\psi}_{AB} = \mathbf{U}_{H\mathbb{1}}\boldsymbol{e}_{00} = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 0 & 1 & 0\\0 & 1 & 0 & 1\\1 & 0 & -1 & 0\\0 & 1 & 0 & -1\end{bmatrix}\begin{bmatrix}1\\0\\0\\0\end{bmatrix} = \begin{bmatrix}\frac{1}{\sqrt{2}}\\0\\\frac{1}{\sqrt{2}}\\0\end{bmatrix}. \tag{2.35b}$$

At this point, there is $|\frac{1}{\sqrt{2}}|^2 = 0.5$ probability of measuring spin (up,up) and spin (down, up), respectively, using the Born rule on the above state. Finally, let us now let the two qubits interfere with the CNOT gate in Eq. (2.32),

$$|\psi'_{AB}\rangle = U_{\text{CNOT}}\,|\psi_{AB}\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}, \tag{2.36a}$$

$$\boldsymbol{\psi}'_{AB} = U_{\text{CNOT}}\boldsymbol{\psi}_{AB} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}. \tag{2.36b}$$

The right hand side state is the equal superposition of (up,up) and (down,down). Upon measurement of one qubit which yields, say, spin up, the wave vector collapses (Postulate 4) to the (up,up) state i.e. measurement immediately changes the state: by measuring one of the qubits we immediately know what the outcome would be if we measured the other qubit. The state in Eq. (2.36) is one of four famous states known as the *Bell states* and the particular one we created is often denoted $|\Phi^+\rangle = \Phi_+$. We can now illustrate the effects of interference by applying a Hadamard gate to both qubits (denoted $U_{HH} = H \otimes H$),

$$U_{HH}\left|\Phi^+\right\rangle = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{2}\left(\frac{2}{\sqrt{2}}\left|00\right\rangle + \cancel{(\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}})\left|01\right\rangle} + \cancel{(\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}})\left|10\right\rangle} + \frac{2}{\sqrt{2}}\left|11\right\rangle\right), \tag{2.37a}$$

$$U_{HH}\Phi_+ = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{2}\left(\frac{2}{\sqrt{2}}\boldsymbol{e}_{00} + \cancel{(\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}})\boldsymbol{e}_{01}} + \cancel{(\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}})\boldsymbol{e}_{10}} + \frac{2}{\sqrt{2}}\boldsymbol{e}_{11}\right). \tag{2.37b}$$

which—due to destructive interference—gets us right back to the original state, that is, $U_{HH}|\Phi^+\rangle = |\Phi^+\rangle$ (or with ML notation: $U_{HH}\Phi_+ = \Phi_+$)

2.2 Noisy Quantum Systems

We learned that a quantum algorithm can be made up from qubits, gates and measurements (Fig. 1.7). To be more specific, this is also known as the *gate model* of quantum computing which is our main focus in this thesis, however, other architectures exist such as adiabatic quantum computation [120], and topological quantum computation [121] also have interesting properties. In fact, many of the quantum circuit architectures we are interested in draws inspiration from adiabatic quantum computation as we shall see in Chapter 3.

Generally, one can divide the types of errors into two categories: systematic (called coherent) and random errors (called incoherent) [122]. An example of coherent errors could be imprecise or finite precision implementation of gates, that is, instead of implementing the $U(\theta)$, our quantum machine systematically makes the operation $U(\theta + c)$ for some (hopefully) small and constant c. On the other hand, incoherent errors can happen due to interaction with the surrounding environment (illustrated in

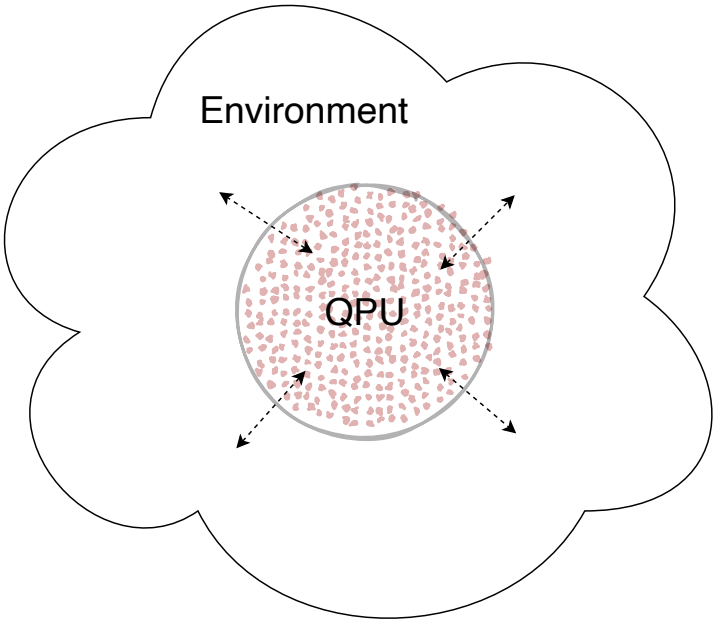


Figure 2.2: Depiction of a quantum processing unit (QPU) interacting with the surrounding environment.

Fig. 2.2), measurement errors, or random gate errors such as gates are implemented with $U(\theta + \epsilon)$ for $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Random errors are typically much more complex processes which depends on the quantum hardware. Furthermore, the source and nature of the errors might even change over time.

While systematic errors can be mitigated [123] or for some quantum algorithms—as we shall see in Chapter 3—under mild assumptions ignored, random errors are less trivial to model and the entire field of quantum error mitigation (introduced in Section 3.5) deals with how to decrease the noise effect on the expectation of observables. Choosing the noise model depends on the physical implementation of the quantum hardware, and in fact, the experimentalist handling our quantum computer would need to characterize the system at hand using e.g. *quantum process tomography* (QPT) [15]. QPT techniques essentially perform a series of measurements such that we can describe the overall noise process of the computer. Together with quantum error correction (QEC), QPT deals with modelling and correcting for random errors happening in the hardware. QPT and QEC are both out of scope for this thesis, and instead, we will outline one of the most commonly used noise models used in the VQA literature: the Pauli Error Model.

2.2.1 Pauli Error Model

One of the most used noise model is the single-qubit Pauli Error Model (PEM) also known as Pauli Channels. In fact, in our simulations we shall exclusively model noise as being discrete one-qubit Pauli noise channels; a simulation strategy often used in VQA research such as in ref. [124].

Not only are Pauli Channels more convenient and easy to simulate, but it has been shown that more general local quantum noise (both coherent and incoherent) can be mapped onto a Pauli channel via randomized compiling [125] or twirling [126]. Moreover, it is possible to experimentally learning the Pauli channel from given quantum hardware using measurements [127]. We shall thus in this thesis only consider noise models which have the following stochastic unitary decomposition

$$\mathcal{E}(\rho) = p_0 \mathbb{1} \rho \mathbb{1}^\dagger + p_1 X \rho X^\dagger + p_2 Y \rho Y^\dagger + p_3 Z \rho Z^\dagger, \tag{2.38a}$$

$$\mathcal{E}(\boldsymbol{\rho}) = p_0 \mathbb{1} \boldsymbol{\rho} \mathbb{1}^\dagger + p_1 X \boldsymbol{\rho} X^\dagger + p_2 Y \boldsymbol{\rho} Y^\dagger + p_3 Z \boldsymbol{\rho} Z^\dagger, \tag{2.38b}$$

where p_i are classical probabilities.

The question of *when* and *which* error happens is thus modelled as a probability of one of the single-

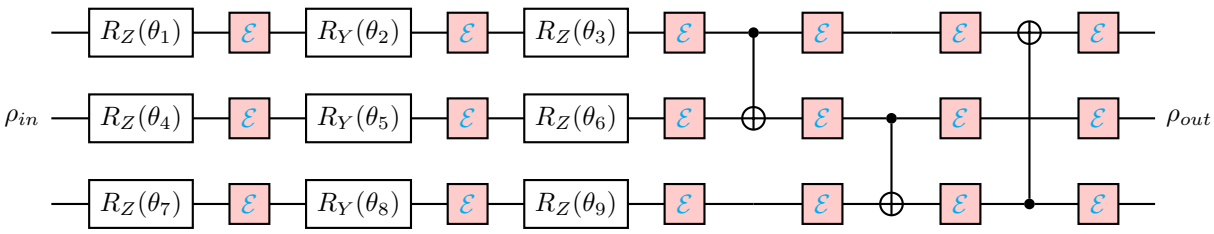


Figure 2.3: Strong Entangling Layer [119] with noise channels after each moment.

qubit Pauli matrices happening after, for example, each *moment* in our circuit. A moment is "a time-slice of operations within a circuit.", that is, operations that happen at the same time [128]. We can now take the quantum circuit in Fig. 2.1 and illustrate our noise model as in Fig. 2.3.

The PEM is widely used but it does not encounter for all types of errors such as the amplitude damping channel. However, in our simulations we will exclusively be dealing with depolarization noise as we shall define momentarily.

Bit-flip As example of a quantum noise channel the bit flip channel has the Kraus operators,

$$K_0 = \sqrt{1-p}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, K_1 = \sqrt{p}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{2.39}$$

where with probability p the bit-flip happens and $1-p$ it does not.

Depolarization Another quantum noise channel example—used extensively in the scientific contributions of this thesis—is the depolarization channel which has the following Kraus operators

$$K_0 = \sqrt{1-p}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, K_1 = \sqrt{p/3}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, K_2 = \sqrt{p/3}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, K_3 = \sqrt{p/3}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.40}$$

corresponding to either a Pauli X,Y or Z gate has randomly been applied with probability p and with probability $1-p$ no random error occurred. The depolarization channel is also equivalent to a convex combination of the original state and the maximally mixed state,

$$D_\lambda(\rho) = (1-\lambda)\rho + \lambda \mathbb{1}, \tag{2.41a}$$

$$D_\lambda(\boldsymbol{\rho}) = (1-\lambda)\boldsymbol{\rho} + \lambda \mathbb{1}, \tag{2.41b}$$

where $\lambda = p$ is the noise level. The depolarization channel is often called the "worst-case scenario" channel [129], since it drags the qubit towards all computational basis states with equal probability; it destroys the coherence (off-diagonals) of the density matrix.

2.2.2 Quantum Channels are generally irreversible

One very important thing happens to the quantum state when acted upon by as CPTP quantum channel of the same dimensionality $\mathcal{E}$ is that information can be lost but not recovered. This is due to the nature of CPTP channels, namely that they are *contractive*, in the sense that any two quantum states ρ_A and ρ_B will be closer to each other / more similar after $\mathcal{E}$ has acted on each state. We mention that there exists more general quantum processes such as not trace-preserving maps where information is gained by measurement, but we leave further explanations to Nielsen and Chuang [15] for the curious reader.

One way to measure how well a specific quantum channel preserves the information, is to ask how close the quantum state is to its previous self, that is, $D(\mathcal{E}(\rho_A), \rho_A)$ using some distance measure D between quantum states. Given that $D(\mathcal{E}(\rho), \rho)$ both depends on $\mathcal{E}$ and ρ, the loss of information is specific to both the state and the quantum channel. An often used similarity measure is the *fidelity*, which is a distance measure between two quantum states ρ_A and ρ_B, defined by,

$$F(\rho_A, \rho_B) := \mathrm{Tr}\left[\sqrt{\sqrt{\rho_A}\rho_B\sqrt{\rho_A}}\right], \tag{2.42a}$$

$$F(\boldsymbol{\rho}_A, \boldsymbol{\rho}_B) := \mathrm{Tr}\left[\sqrt{\sqrt{\boldsymbol{\rho}_A}\boldsymbol{\rho}_B\sqrt{\boldsymbol{\rho}_A}}\right]. \tag{2.42b}$$

It is easy to verify that for two different quantum states ρ_A and ρ_B, their fidelity is the same or have increased after an error channel has been applied, that is, $F(\mathcal{E}(\rho_A), \mathcal{E}(\rho_B)) \geq F(\rho_A, \rho_B)$ [15].

2.3 Spin systems

As mentioned in the introduction, a key application of quantum computers is to simulate quantum systems and it turns out that many interesting machine learning problems can be mapped into problems of quantum simulation. Just as simulations of how the planets move around the sun can be simulated with an orrery, we can use a quantum computer to simulate a quantum system. To get there, we introduce the following statistical mechanics which shall constitute the foundation on which we introduce variational quantum algorithms in Chapter 3.

In the introduction, we mentioned that the spin of an electron is related to magnetic moment and in this section we will think of the spin as a tiny three dimensional magnet. The magnet can be in any superposition of up and down but for now let us start putting all electrons either spin up or spin down, or similar for magnets, magnetic north pointing either up or down, respectively. We imagine taking three such magnets, placing them next to each other and putting a barrier between nearest neighbor magnets (1,2) and (2,3), as depicted in Fig. 2.4. The barriers are parameterized with a number $J_{i,j}$ indicating the interaction strength between neighbor magnets i and j: a large $J_{i,j}$ means a lot of interaction and vice versa. We can think of $1/J_{i,j}$ as being the width of the wall seen in Fig. 2.4. If there is a thin wall, magnet 1 would naturally tend to flip in order to anti-align with magnet 2; in other words it takes more *energy* to keep or put the magnetic north pointing up and less energy if they are anti-aligned. As the barrier width increase ($J_{1,2}$ decrease) this force will decrease and it would take more energy to flip the first magnet. We could also apply an external strong magnetic field to each site which—if strong enough—would make the effects of interaction between the small magnets negligible. The next paragraph concerns modelling this behavior, that is, mathematically describe the energy with a function $\mathcal{H}$ of such system in order to find magnet configurations (how they point) that lower the energy. The energy function $\mathcal{H}$ is also called the *Hamiltonian* or *energy operator* of the system, and it is a function what state the magnets are in. As we shall see throughout the thesis, the concept of the Hamiltonians will be used continuously.

The Ising model We model the energy of the magnets in Fig. 2.4 using the Ising model [130]. Hence for a specific sequence of ups and downs, we can plug it into the Hamiltonian and get a scalar value. When the spins are in the quantum state of up and down given by a state vector or density matrix, we need to use the formalism of quantum mechanics to compute expectation values of the energy. As we know from Postulate 4, observables—such as the energy of a system of spins—is associated with a Hermitian

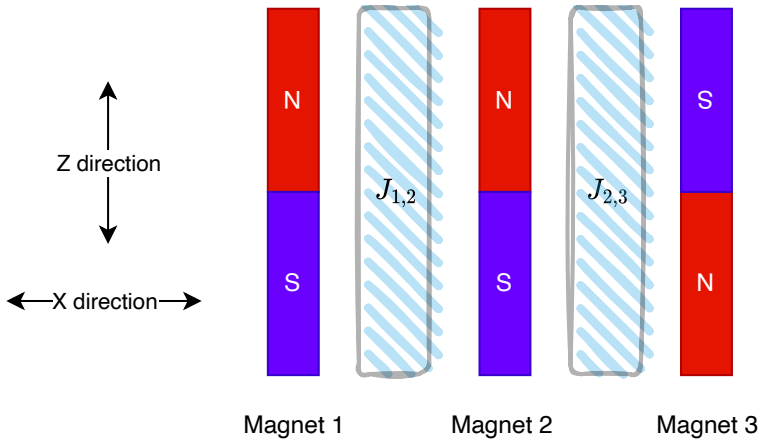


Figure 2.4: Three magnets aligned with a barrier controlling their interaction strength. From the Ising model in Eq. (3.7) the pairwise interaction strength of $J_{i,j}$ indicates the interaction strength between neighbor magnets i and j.

operator. For the particular system in Fig. 2.4 the Hermitian operator can be written as a sum of two contributions: one from the interaction between the spins/magnets and one from the external magnetic field,

$$\mathcal{H}_I = -\underbrace{\sum_i b_i Z_i}_{\text{external}} - \underbrace{\sum_{<i,j>} J_{i,j} Z_i Z_j}_{\text{interaction}}, \tag{2.43}$$

where Z_i is the spin up/down operator for the i'th qubit constructed using the following tensor product,

$$Z_i = \overbrace{\mathbb{1}_2 \otimes \mathbb{1}_2 \otimes \ldots \mathbb{1}_2}^{i-1} \otimes Z \otimes \overbrace{\mathbb{1}_2 \otimes \ldots \otimes \mathbb{1}_2}^{N-i}, \tag{2.44}$$

and Z is the Pauli-Z matrix from Eq. (2.22). The energy expectation of N spins in a pure state $|\psi\rangle = \boldsymbol{\psi}$ is thus

$$\langle \mathcal{H}_I \rangle = \langle \psi | \mathcal{H}_I | \psi \rangle , \tag{2.45a}$$

$$\langle \mathcal{H}_I \rangle = \boldsymbol{\psi}^\dagger \mathcal{H}_I \boldsymbol{\psi}. \tag{2.45b}$$

We also learned from Postulate 4 that when we measure the state $|\psi\rangle = \boldsymbol{\psi}$, it collapses into an eigenstate of the measurement operator $\mathcal{H}_I$. The outcome value is the corresponding energy eigenvalue.

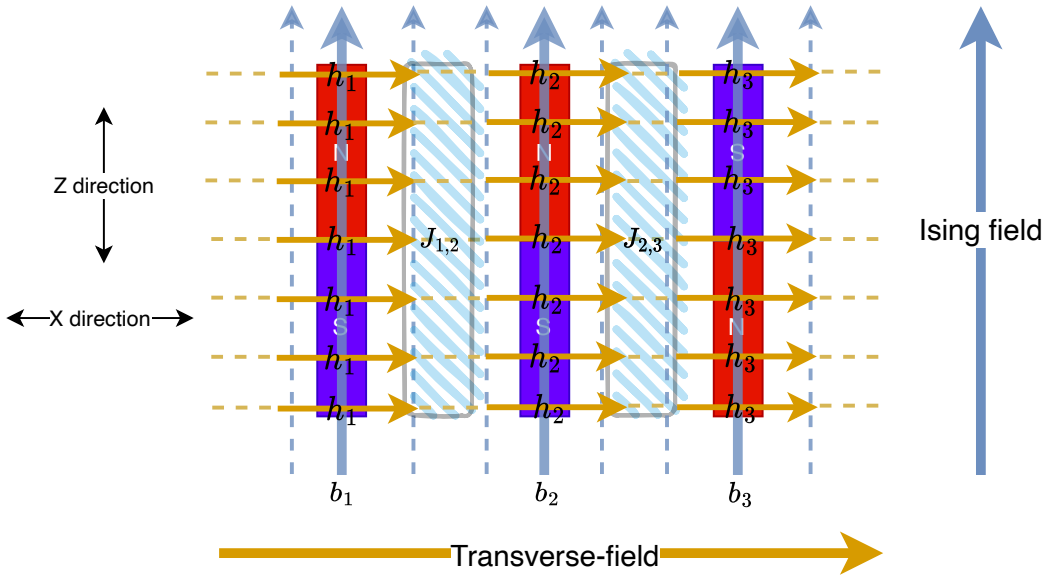


Figure 2.5: Three magnets with a barrier controlling their interaction strength and two orthogonal external magnetic fields (Ising field in the Z direction and Transverse field in the X direction). This system is modelled the energy Hamiltonian in Eq. (3.8).

The set of eigenvalues of an operator is also called the *spectrum* of the operator, and what Nature tries to do via the principle of least action [131], is to put the system in the eigenstate corresponding to the minimum eigenvalue, that is, the *groundstate*. However, this is hard due to many local minima. For two-state magnets with no barriers or external field, the groundstate would be alternating magnets (note that there would be two groundstates). But for more general systems of spins which can be in any superposition of up/down, the energy function describing the situation in Fig. 2.4 with an external magnetic field in the up/down direction is given by the Ising model in Eq. (3.7). It turns out that the minimum energy eigenstate to Eq. (3.7) is a "classical" spin state, that is, one of the computational basis states. This is because $\mathcal{H}_I$ is made up from diagonal matrices (identities and Pauli Z) and hence it is already written in its spectral decomposition with eigenvectors corresponding to all the computational basis states. There exists more exotic Hamiltonians with eigenstates being quantum superposition states, and one example is the Transverse-field Ising model [132]. The extension from the Ising model is simple: we apply another magnetic field with strength h_i at each spin/magnet but this time in the X-direction (see Fig. 2.5). The updated Hamiltonian is then

$$\mathcal{H}_{TFI} = -\sum_i b_i Z_i - \sum_{<i,j>} J_{i,j} Z_i Z_j - \sum_i h_i X_i, \tag{2.46}$$

where X_i is the Pauli X observable on qubit i. The eigenvectors of X are given by

$$\left|\lambda_x^{(1)}\right\rangle = \frac{1}{\sqrt{2}}\left|0\right\rangle + \frac{1}{\sqrt{2}}\left|1\right\rangle, \left|\lambda_x^{(2)}\right\rangle = \frac{1}{\sqrt{2}}\left|0\right\rangle - \frac{1}{\sqrt{2}}\left|1\right\rangle, \tag{2.47a}$$

$$\boldsymbol{\lambda}_x^{(1)} = \frac{1}{\sqrt{2}}\boldsymbol{e}_0 + \frac{1}{\sqrt{2}}\boldsymbol{e}_1, \boldsymbol{\lambda}_x^{(2)} = \frac{1}{\sqrt{2}}\boldsymbol{e}_0 - \frac{1}{\sqrt{2}}\boldsymbol{e}_1, \tag{2.47b}$$

which are equal superposition states of up/down. Thus depending on the coefficients of the magnetic fields (b_i,$J_{i,j}$,h_i) the X terms would drag the eigenvectors towards superpositions whereas the Z terms push towards classical eigenstates. There exists even more exotic Hamiltonians with multiple interaction terms in various directions modelling different types of systems. The most general Hamiltonian containing all Pauli interactions up to pairs of two qubits (called 2-local systems) can be written on the form

$$\mathcal{H}_{general} = -\sum_i a_i X_i - \sum_i b_i Y_i - \sum_i c_i Z_i - \sum_{<i,j>} L_{i,j} X_i X_j - \sum_{<i,j>} K_{i,j} Y_i Y_j - \sum_{<i,j>} J_{i,j} Z_i Z_j. \tag{2.48}$$

Finding the groundstate of arbitrary Hamiltonians is difficult [133]. For N spins, the Hamiltonian is a $2^N \times 2^N$ matrix and storing this matrix alone on a classical laptop is difficult for N not much bigger than 20. However, quantum computers naturally contains spins and estimating ground state energies is thus of high relevance which is indeed what the next chapter concerns. Furthermore, if one can encode a specific problem into the form of Eq. (2.48), such that the groundstate is a useful state then a quantum computer might be useful in assisting with such estimation. It turns out a lot of problems can be encoded onto a energy minimization problem and this is the cornerstone of variational quantum algorithms as described in Chapter 3.

Experimentally, for a quantum computer the expectation for general Hamiltonians (see Eq. (3.3)) might contain measurements not directly implemented on our quantum computer. For example, many quantum computers only measure spin variables in the computational basis (our up/down). Thus in order to measure in the X or Y basis, a gate operation applied to the qubit(s) needs to be applied such that the state is projected onto the new basis. We can apply the inverse phase gate ($S^\dagger$) followed by a Hadamard gate or only the Hadamard just prior to measurement in order to measure in the Y or X basis, respectively. As such, general expectations are computed by estimating each term via measuring in the

corresponding Pauli Basis and subsequently insert the estimates in Eq. (2.48).

2.3.1 The adiabatic theorem

We just saw that Hamiltonians can be both "simple" and "complex" in terms of how easy it is to put spins in the groundstate. If we have a trivial Hamiltonian $\mathcal{H}_0$ and a complex Hamiltonian $\mathcal{H}_1$, we can define a new time-dependent Hamiltonian for $t \in [0, 1]$,

$$\mathcal{H}(t) = t\mathcal{H}_1 + (1-t)\mathcal{H}_0. \tag{2.49}$$

The adiabatic theorem [134] tells us, that if a quantum system in state ρ starts in the ground state of $\mathcal{H}_0$ and is evolved under Eq. (2.49) starting with $t = 0$, then slowly let $t \to 1$, the system ρ will end up in the ground state of H_1 [135]. The speed limit of which $t \to 1$ is determined by the so-called spectral *gap* denoted $g^2(t)$ of $\mathcal{H}(t)$ which is the energy distance between the ground state and first excited state energies, and that it must scale according to $\frac{1}{\min_t g^2(t)}$.

One can discretize the adiabatic process, such that the adiabatic process is taken in steps of size Δt which is known as *Trotterization* [136, 137],

$$\begin{aligned}
\mathcal{H}(0) &= \mathcal{H}_0, \\
\mathcal{H}(\Delta t) &= \Delta t\mathcal{H}_1 + (1-\Delta t)\mathcal{H}_0, \\
\mathcal{H}(2\Delta t) &= 2\Delta t\mathcal{H}_1 + (1-2\Delta t)\mathcal{H}_0, \\
&\vdots \\
\mathcal{H}(1-\Delta t) &= (1-\Delta t)\mathcal{H}_1 + \Delta t\mathcal{H}_0, \\
\mathcal{H}(1) &= \mathcal{H}_1,
\end{aligned}$$

where the smaller the step size Δt, the more steps needs to be taken and the closer to the analog adiabatic path the process will be. Adiabatic quantum computing (AQC) [138], such as the D-Wave quantum annealer [139], is exactly about exploiting this fact. At its core, AQC encodes a problem in the groundstate of a *problem/cost* Hamiltonian $\mathcal{H}_1$, starts the qubits in the groundstate of a *initial/mixer* Hamiltonian

$\mathcal{H}_0$, applies the magnetic field in Eq. (2.49) starting with $t = 0$ and then slowly let $t \to 1$ according to the spectral gap. It should be, however, noted that the gap unfortunately can be exponentially small [140] essentially meaning generally no exponential speedups for NP-hard problems. As we shall see in Chapter 3, one of the key ideas for NISQ algorithms uses Trotterization to build a specific quantum circuit.

2.4 The Boltzmann Distribution

Let us briefly consider a system of N classical magnets which can not be in any superposition state; only 1 out of 2^N states. Each outcome n has an associated energy E_n. If we know the probability p_n of magnets being in the n'th outcome—say, if someone prepared it for us—we can write the average (expectation value) energy as

$$\langle E \rangle = \sum_n E_n p_n. \tag{2.50}$$

Associated with that distribution is the Shannon entropy (see Eq. (2.14))

$$S(p_n) = -\sum_n p_n \log p_n. \tag{2.51}$$

We now want to find the probability distribution $p(n)$ according to the maximal entropy principle [141, 142] under two constraints,

$$\sum_n p_n = 1 \iff \sum_n p_n - 1 = 0, \tag{2.52}$$

$$\langle E \rangle = \sum_n E_n p_n \iff \sum_n E_n p_n - \langle E \rangle = 0. \tag{2.53}$$

Instead of maximizing $S(p_n)$ we will minimize $-S(p_n)$, again given the constraints. To do constrained optimization, we define a new function G using Lagrange multipliers λ_1 and λ_2,

$$G = \lambda_1 \left(\sum_n p_n - 1 \right) + \lambda_2 \left(\sum_n E_n p_n - \langle E \rangle \right) - S(p_n), \tag{2.54}$$

and do unconstrained optimization to find the minimum of G w.r.t. some p_j. By differentiating Eq. (2.54) and setting it equal to zero, we get

$$\begin{aligned}\frac{\partial G}{\partial p_j} &= \frac{\partial}{\partial p_j}\left[\lambda_1\left(\sum_n p_n - 1\right) + \lambda_2\left(\sum_n E_n p_n - \langle E\rangle\right) + \sum_n p_n \log p_n\right] = 0,\\ &= \frac{\partial}{\partial p_j}\left[\lambda_1 p_j + \lambda_2 E_j p_j + p_j \log p_j\right] = 0,\\ &= \lambda_1 + \lambda_2 E_j + \log p_j + 1 = 0.\end{aligned}$$

Isolating the probability distribution yields,

$$\log p_j = -(\lambda_1 + 1) - \lambda_2 E_j \iff p_j = e^{-(\lambda_1+1)} e^{-\lambda_2 E_j}.$$

Given that $e^{-(\lambda_1+1)}$ is just some constant, we can define $Z := e^{(\lambda_1+1)}$ and not gain/loose anything. Same thing for $\beta := \lambda_2$. Thus we have derived a specific probability distribution known as the *Boltzmann* distribution,

$$p_n = \frac{1}{Z} e^{-\beta E_n}. \tag{2.55}$$

The next question is what Z and β are. The first constraint gives us Z,

$$\sum_n p_n = \frac{1}{Z}\sum_n e^{-\beta E_n} = 1 \iff Z = \sum_n e^{-\beta E_n}. \tag{2.56}$$

We call Z the *partition function* and it is a function of β: $Z(\beta) = \sum_n e^{-\beta E_n}$, when the energy levels E_n are fixed. The second constraint gives,

$$\sum_n p_n E_n = \sum_n \underbrace{\frac{1}{Z} e^{-\beta E_n}}_{p_n} E_n = \langle E\rangle. \tag{2.57}$$

Hence whatever β is, it determines the average energy and vice versa: high β minimizes the average energy and small β increases the average energy. Differentiating the partition function w.r.t. β and subsequently divide with minus the partition function itself, we get

$$-\frac{1}{Z}\frac{\partial Z}{\partial \beta} = \sum_n \underbrace{\frac{1}{Z}e^{-\beta E_n}}_{p_n} E_n = \langle E \rangle \,. \tag{2.58}$$

This is also equal to $\langle E \rangle = -\frac{1}{Z}\frac{\partial Z}{\partial \beta} = -\frac{\partial \log Z}{\partial \beta}$. We will see shortly, that β is the inverse temperature, $\beta = \frac{1}{T}$, but first we will rewrite the entropy.

Rewriting the entropy Now that we know p_n, we can plug this in to the definition of entropy

$$\begin{aligned}
S &= -\sum_n p_n \log p_n, \\
&= -\sum_n \frac{1}{Z}e^{-\beta E_n} \log\left[\frac{1}{Z}e^{-\beta E_n}\right], && \text{(insert Boltzmann dist)} \\
&= -\sum_n \frac{1}{Z}e^{-\beta E_n}\left[-\log Z - \beta E_n\right], && \text{(rule of logarithm)} \\
&= \sum_n \frac{1}{Z}e^{-\beta E_n}\left[\log Z + \beta E_n\right], && \text{(cancel minus)} \\
&= \sum_n \frac{1}{Z}e^{-\beta E_n} \log Z + \sum_n \frac{1}{Z}e^{-\beta E_n}\beta E_n, && \text{(expand product)} \\
&= \frac{1}{Z}\underbrace{\sum_n e^{-\beta E_n}}_{Z} \log Z + \beta \langle E \rangle, && \text{(substitute def. of } Z\text{)} \\
&= \log Z + \beta \langle E \rangle \,.
\end{aligned}$$

The term $\log Z$ is related to the *free energy*, in fact, the free energy is defined by $F := -\frac{\log Z}{\beta}$, which by rearranging the above derivation yields

$$F = H - \frac{1}{\beta}S, \tag{2.59}$$

also known as the *Gibbs* free energy. We shall use the free energy as loss function in paper A in order to approximate a *thermal states* which is a specific state a system can obtain for example when getting in contact with an environment / heat bath.

The inverse temperature If we differentiate the entropy, we get

$$
\begin{aligned}
dS &= \beta d\langle E\rangle + \langle E\rangle d\beta + d\log Z \\
&= \frac{\partial \log Z}{\partial \beta}\partial\beta + \langle E\rangle \partial\beta + \beta d\langle E\rangle \\
&= \partial\beta \left(\underbrace{\frac{\partial \log Z}{\partial \beta}}_{-\langle E\rangle} + \langle E\rangle \right) + \beta d\langle E\rangle \\
&= \beta d\langle E\rangle
\end{aligned}
\tag{2.60}
$$

which is the definition of inverse temperature:

$$
\beta = \frac{dS}{d\langle E\rangle} = \frac{1}{T} \tag{2.61}
$$

In Eq. (2.61), we see that β essentially describes how the entropy (chaos/unpredictability) of the Boltzmann distribution changes when we change the average energy of the system.

2.5 Thermal States

When we looked at the Hamiltonian for a set of spins Eq. (2.48) this was in fact for zero temperature systems. If we take the spins a bring them in weak interaction with the surrounding environment—also called a heat bath— the system does not naturally tend towards the lowest energy state. Instead, the system approaches the *thermal state*, which is a state that does not minimize the energy but the free energy (Eq. (2.59)). The same thing happens to a hot cup of coffee if we leave it for some time at a room: it will approach equilibrium and get room temperature. We say that the system *thermalize* or reaches *thermal equilibrium*. Once the state of spins has found the global minimum of the free energy its energies has a particular form we just encountered: the Boltzmann distribution. The thermal state is thus given as

$$
\rho = \frac{1}{Z} e^{-\beta \mathcal{H}}, \tag{2.62}
$$

where $\mathcal{H}$ is the energy operator for the system (such as the Ising model) and β is the effective inverse temperature that comes both from the thermometer but also from all the noisy interactions between

system and environment. For $\beta \to \infty$, i.e. for $T \to 0$, the free energy $F \to H$ (see Eq. (2.59)) and thus ρ approaches the groundstate of H as this is the minimum (free) energy state. However if we pull out samples from Eq. (2.62) for $T > 0$, that is, measuring the spin of each qubit, each spin configuration comes with a certain probability of being observed. That probability, as we can see from Eq. (2.62), follows a Boltzmann distribution with the minimum energy eigenstate being the most likely one to sample, the first energy eigenstate being second most likely to sample, etc. As the temperature T increases, all states approaches the same probability of being observed, that is, ρ approaches the maximally mixed state (Eq. (2.10)). Many classical machine learning algorithms contain subroutines which samples from distributions such as the Boltzmann distribution, i.e., states such as these thermal states. If one is able to prepare such state with a quantum computer, we have a "natural" distribution to sample from as compared to numerical approximations for computing gradients of Restricted Boltzmann Machines.

2.6 Probabilistic Machine Learning

We spend some effort formalizing quantum mechanics with focus on quantum computing theory, noise models and some of the underlying statistical mechanics. In this section, we formalize machine learning concepts relevant to the scientific contributions of the thesis. In particular, we introduce the general objective of *learning* and tools of the field relevant to all Papers. We subsequently derive Gaussian processes which is the foundation for papers C and D. Finally, we go over the Restricted Boltzmann Machine which is relevant for Paper A.

Machine learning (ML) is a sub discipline in artificial intelligence (AI) focusing on the mathematical, algorithmic and statistical aspects of modelling data [143]. The key task in machine learning is for the computer to be able to learn without being explicitly programmed [144]. "Learning" here refers to a process in which a *model* fits to a dataset such that a predefined loss function is minimized. As we shall see in the next chapters, many of the ideas for NISQ algorithms come back to such a learning task. We therefore spend some effort outlining key concepts which applies to all scientific contributions.

We take starting point in a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ which is a collection of n samples of pairs of inputs $\mathbf{x}_i = [1, x_1, ..., x_D]^\top$ and output scalar value y_i. For example, $\mathbf{x}_i$ could contain weight, height and age and y_i could be blood pressure. We also refer to the collection of all data input vectors as the input data matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]$, and the collection of the corresponding outputs as the output data vector

$\mathbf{y} = [y_1, y_2, ..., y_n]^\top$. The goal is to find a function f that maps $\mathbf{x}_i \to y_i$. We let the model *train*/fit on a subset of our data $\mathcal{D}$, and then we test its performance on the remaining data to get an estimate for the generalization error, that is, the average error the model makes on new unseen data [ref]. Let us initially focus on a model $f_\mathbf{w}(\mathbf{x})$ with parameters $\mathbf{w}$ that takes the input data vector $\mathbf{x}$ and outputs a scalar $\hat{y}$ using a linear function given by

$$\hat{y} = f_\mathbf{w}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + ... + w_D x_D. \tag{2.63}$$

The goal of *learning*/training is to estimate a good set of weights $\mathbf{w} = [w_0, w_1, ..., w_D]^\top$ using the training set such that when a new unseen test datapoint $\mathbf{x}_*$ enters the model, the corresponding prediction $\hat{y}_*$ is close to the actual output value, i.e., $\hat{y}_* \approx y_*$. All supervised machine learning models follow this recipe, but the model—the input/output function f—might be linear, a polynomial, a neural network or something different.

We can expand the type of model in Eq. (2.63) from a deterministic one to a stochastic one by modelling an *uncertainty* on the output y. Instead of only providing one point estimate of the value $\hat{y}_i$ our model outputs a distribution $p(\hat{y}_i|\mathbf{x}_i, \mathbf{w})$. The uncertainty over $\hat{y}$ might be because the model itself is uncertain due to limited data (known as *epidemistic* uncertainty) or it might be because the underlying data is contaminated with noise (known as *aleatoric* uncertainty) or a combination. A common approach is to model the data with normal distributed noise around the prediction in Eq. (2.63),

$$p(\hat{y}_i|\mathbf{x}_i, \mathbf{w}) = \mathcal{N}(\mathbf{w}^\top \mathbf{x}_i, \sigma^2).$$

A normal distribution is meaningful when multiple error sources adds to the sampled output due to the central limit theorem [145]. Given n training points—which we assume to be independent and identically distribution (i.i.d.)—we can compute the likelihood function which can be seen as a goodness-of-fit measure,

$$L := p(\mathcal{D}|\mathbf{w}) = \prod_{i=1}^{n} p(y_i|\mathbf{x}_i, \mathbf{w}) = \prod_{i=1}^{n} \mathcal{N}(\mathbf{w}^\top \mathbf{x}, \sigma^2), \tag{2.64}$$

i.e. the better the weights $\mathbf{w}$ fit the data, the more "likely" the data will be under the model with parameters $\mathbf{w}$ will be. This is because the term $\mathcal{N}(\mathbf{w}^\top \mathbf{x}, \sigma^2)$ will be numerically larger and thus we end up multiplying together larger numbers yielding a larger likelihood. Finding $\mathbf{w}$ is often found using one of

three following techniques. First approach is to maximize the likelihood function. That is, the maximum likelihood coefficients $\mathbf{w}_{\text{MLE}}$ are given by

$$\mathbf{w}_{\text{MLE}} = \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{i=1}^{n} p(y_i, \mathbf{x}_i|\mathbf{w}). \tag{2.65}$$

The second technique uses a prior probability distribution $p(\mathbf{w})$ over the weights. This distribution is often Gaussian $p(\mathbf{w}) = \mathcal{N}(\mathbf{0}, \sigma \cdot \mathbb{1})$. We now have two variables, each with a distribution: the data $\mathcal{D}$ and the weights $\mathbf{w}$, thus we can use Bayes theorem which connects two or more random variables in the following relation,

$$p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}. \tag{2.66}$$

A set of weights $\mathbf{w}_{\text{MAP}}$ can be found by maximizing this posterior, and the solution is referred to as maximum *a posteriori* (MAP) solution given by,

$$\mathbf{w}_{\text{MAP}} = \underset{\mathbf{w}}{\operatorname{argmax}} \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})} = \underset{\mathbf{w}}{\operatorname{argmax}}\, p(D|\mathbf{w})p(\mathbf{w}). \tag{2.67}$$

If we not only want a point estimate of the weights but in fact a probability distribution over them, we can use Bayes theorem once again to get Bayes estimate,

$$\mathbf{w}_{\text{BE}} = \int \mathbf{w} p(\mathbf{w}|\mathcal{D}) \mathrm{d}\mathbf{w}. \tag{2.68}$$

If everything is assumed to be normally distributed then $\mathbf{w}_{\text{BE}} = \mathbf{w}_{\text{MAP}}$ given that the mean of a normal distribution is also where the distribution has its maximum, but in general, this is not the case. If we plug the Gaussian likelihood (Eq. (2.64)) and prior into Bayes theorem in order to obtain the posterior over the weights we get the *bayesian linear regression* solution,

$$\begin{aligned} p(\mathbf{w}|\mathbf{X}, \mathbf{y}) &\propto \underbrace{\exp\left\{-\frac{1}{2\sigma_n^2}(\mathbf{y} - \mathbf{X}^\top \mathbf{w})^\top (\mathbf{y} - \mathbf{X}^\top \mathbf{w})\right\}}_{\text{likelihood}} \\ &\underbrace{\exp\left\{-\frac{1}{2}\mathbf{w}^\top \boldsymbol{\Sigma}_p^{-1} \mathbf{w}\right\}}_{\text{prior}}, \\ &\propto \exp\left\{(\mathbf{w} - \bar{\mathbf{w}})^\top \mathbf{A}^{-1} (\mathbf{w} - \bar{\mathbf{w}})\right\}, \end{aligned} \tag{2.69}$$

where $\bar{\mathbf{w}} = \frac{1}{\sigma_n^2}\mathbf{A}^{-1}\mathbf{X}\mathbf{y}$ for $\mathbf{A} = \frac{1}{\sigma_n^2}\mathbf{X}\mathbf{X}^\top + \boldsymbol{\Sigma}_p$. We here see that the numerator is a also a Gaussian! It is therefore possible to write in a more compact form

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}) \sim \mathcal{N}(\bar{\mathbf{w}}, \mathbf{A}^{-1}). \tag{2.70}$$

From this posterior it is possible to provide the predictive distribution over y. That is, given the training set $\mathbf{X}, \mathbf{y}$ and a test input $\mathbf{x}_*$, we can write,

$$\begin{aligned} p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) &= \int p(y_*, \mathbf{w}|\mathbf{x}_*, \mathbf{X}, \mathbf{y})\mathrm{d}\mathbf{w}, \\ &= \int p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}, \mathbf{w})p(\mathbf{w}|\mathbf{x}_*, \mathbf{X}, \mathbf{y})\mathrm{d}\mathbf{w}, && \text{(split joint distribution)} \\ &= \int p(y_*|\mathbf{x}_*, \mathbf{w})\underbrace{p(\mathbf{w}|\mathbf{X}, \mathbf{y})}_{\text{posterior}}\mathrm{d}\mathbf{w}. && \text{(assume } \mathbf{w} \text{ independent of } \mathbf{x}_*) \end{aligned}$$

In general this integral is intractable unless we assume likelihood $p(\mathbf{y}, \mathbf{X}|\mathbf{w})$ and the prior $p(\mathbf{w})$ are Gaussian, which we, in fact, do for now. The predictive distribution thus becomes

$$p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}\left(\bar{\mathbf{w}}^\top\mathbf{x}_*, \mathbf{x}_*^\top\frac{1}{\sigma_n^2}\mathbf{X}\mathbf{X}^\top + \boldsymbol{\Sigma}_p^{-1}\mathbf{x}_*\right). \tag{2.71}$$

We note here that the mean in Bayesian linear regression is exactly the ordinary least square solution [146].

2.7 Gaussian Processes

The Gaussian Process (GP) [147] can be seen as a generalization of Bayesian regression to go beyond a linear predictive distribution. Once we have established bayesian linear regression, the idea is sime: <u>first</u> project the inputs into some high dimensional space and <u>then</u> apply the linear model in this space instead of directly on the inputs themselves. The projection can be in many different ways. For example, for a scalar x, we can project it via

$$\boldsymbol{\phi}(x) = [1, x, x^2, \sin(x), ...]^\top. \tag{2.72}$$

How to choose this projection map? It turns out we don't have to specify this specific map, but instead have to specify something else known as the *kernel function*. In order to show this, we will first assume the project is given: given a function $\boldsymbol{\phi} : \mathbb{R}^D \to \mathbb{R}^N$ then

$$y = \underbrace{\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x})}_{f(\mathbf{x})} + \epsilon \tag{2.73}$$

for $\mathbf{w} \in \mathbb{R}^L$. Replace $\mathbf{x}$ with $\boldsymbol{\phi}(\mathbf{x})$:

$$\begin{aligned} p(y_*|\mathbf{x}_*, \mathbf{x}, \mathbf{y}) &= \mathcal{N}\left(\frac{1}{\sigma_n^2}\mathbf{x}_*^\top \mathbf{A}^{-1}\mathbf{X}\mathbf{y}, \mathbf{x}_*^\top \mathbf{A}^{-1}\mathbf{x}_*\right) \\ &\Rightarrow \mathcal{N}\left(\frac{1}{\sigma_n^2}\boldsymbol{\phi}(\mathbf{x}_*)^\top \mathbf{A}^{-1}\boldsymbol{\phi}(\mathbf{X})\mathbf{y}, \boldsymbol{\phi}(\mathbf{x}_*)^\top \mathbf{A}^{-1}\boldsymbol{\phi}(\mathbf{x}_*)\right) \end{aligned} \tag{2.74}$$

where $\mathbf{A} = \frac{1}{\sigma_n^2}\boldsymbol{\phi}(\mathbf{X})\boldsymbol{\phi}(\mathbf{X})^\top + \boldsymbol{\Sigma}_p^{-1}$. The only problem here is that generally inverting a matrix has complexity cubed in the dimension $\mathcal{O}(L^3)$, hence if feature dimension L is large we are in trouble. It turns out we can rewrite this to be in the dimension of the number of samples rather than the feature space. Using the notation

$$\begin{aligned} \boldsymbol{\phi} &:= \boldsymbol{\phi}(\mathbf{x}) \\ \boldsymbol{\phi}_* &:= \boldsymbol{\phi}(\mathbf{x}_*) \\ \Phi &:= \boldsymbol{\phi}(\mathbf{X}) \\ \mathbf{K} &:= \Phi^\top \boldsymbol{\Sigma}_p \Phi \\ \mathbf{A} &:= \frac{1}{\sigma_n^2}\Phi\Phi^\top + \boldsymbol{\Sigma}_p^{-1} \end{aligned} \tag{2.75}$$

it is possible to rewriting the mean vector via:

$$\begin{aligned}
\frac{1}{\sigma_n^2}\Phi(\mathbf{K}+\sigma_n^2\mathbb{1}) &= \frac{1}{\sigma_n^2}\Phi(\Phi^\top\boldsymbol{\Sigma}_p\Phi+\sigma_n^2\mathbb{1}) \\
&= \frac{1}{\sigma_n^2}\Phi\Phi^\top\boldsymbol{\Sigma}_p\Phi+\Phi\mathbb{1} \\
&= (\mathbf{A}-\boldsymbol{\Sigma}_p^{-1})\boldsymbol{\Sigma}_p\Phi+\Phi I \\
&= (\mathbf{A}\boldsymbol{\Sigma}_p-\mathbb{1})\Phi+\Phi\mathbb{1} \\
&= \mathbf{A}\boldsymbol{\Sigma}_p\Phi \\
&\iff \\
\mathbf{A}^{-1}\frac{1}{\sigma_n^2}\Phi(\mathbf{K}+\sigma_n^2\mathbb{1})(\mathbf{K}+\sigma_n^2\mathbb{1})^{-1} &= \mathbf{A}^{-1}\mathbf{A}\boldsymbol{\Sigma}_p\Phi(\mathbf{K}+\sigma_n^2\mathbb{1})^{-1}
\end{aligned}$$

yielding

$$\mu_{\mathbf{y}_*|\mathbf{x}_*,\mathbf{X},\mathbf{y}} = \frac{1}{\sigma_n^2}\boldsymbol{\phi}_*^\top\mathbf{A}^{-1}\Phi\mathbf{y} = \boldsymbol{\phi}_*^\top\underbrace{\boldsymbol{\Sigma}_p\Phi(K+\sigma_n^2 I)^{-1}}_{\frac{1}{\sigma_n^2}\mathbf{A}^{-1}\Phi}\mathbf{y} \tag{2.76}$$

Similarly, one can rewrite the covariance matrix to depend on the data dimension rather than the feature dimension [147]:

$$\boldsymbol{\Sigma}_{y_*|\mathbf{y}} = \mathbf{x}_*^\top\frac{1}{\sigma_n^2}\mathbf{X}\mathbf{X}^\top+\boldsymbol{\Sigma}_p^{-1}\mathbf{x}_* = \boldsymbol{\phi}_*^\top\boldsymbol{\Sigma}_p\boldsymbol{\phi}_* - \boldsymbol{\phi}_*^\top\boldsymbol{\Sigma}_p\Phi(\Phi^\top\boldsymbol{\Sigma}\Phi+\sigma_n^2 I)^{-1}\Phi^\top\boldsymbol{\Sigma}_p\boldsymbol{\phi}_* \tag{2.77}$$

Kernel Trick We note a specific re-occurring structure in the predictive mean and variance

$$\begin{aligned}
\boldsymbol{\mu}_{y_*|\mathbf{y}} &= \overbrace{\boldsymbol{\phi}_*^\top\boldsymbol{\Sigma}_p\Phi}(\overbrace{\Phi^\top\boldsymbol{\Sigma}\Phi}+\sigma_n^2 I)^{-1}\mathbf{y}, \\
\boldsymbol{\Sigma}_{y_*|\mathbf{y}} &= \underbrace{\boldsymbol{\phi}_*^\top\boldsymbol{\Sigma}_p\boldsymbol{\phi}_*} - \underbrace{\boldsymbol{\phi}_*^\top\boldsymbol{\Sigma}_p\Phi}(\underbrace{\Phi^\top\boldsymbol{\Sigma}\Phi}+\sigma_n^2 I)^{-1}\underbrace{\Phi^\top\boldsymbol{\Sigma}_p\boldsymbol{\phi}_*}
\end{aligned}$$

namely the feature space is always on the form of an inner product (scalar)

$$\boldsymbol{\phi}(\mathbf{x})^\top\boldsymbol{\Sigma}_p\boldsymbol{\phi}(\mathbf{x}')$$

where $\mathbf{x}$ and $\mathbf{x}'$ are either training or tests points. Let us define the following function

$$k(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\phi}(\mathbf{x}_i)^\top \boldsymbol{\Sigma}_p \boldsymbol{\phi}(\mathbf{x}_j) = \psi(\mathbf{x}_i)^\top \psi(\mathbf{x}_j) = k_{ij},$$

and call it a *kernel*. We note that a kernel can be seen as similarity measure between pairs of vectors. Given the structure seen in the predictive mean and covariance, we can apply the so-called "kernel trick" (see [52] for more details on the for kernel criteria), which is

If an algorithm is defined solely in terms of inner products in input space then it can be lifted into feature space by replacing occurrences of those inner products by $k(\mathbf{x}, \mathbf{x}')$.

Kernels represent the data only through a set of pairwise similarity comparisons between the original data observations x (in the lower dimensional space), instead of explicitly applying the transformations. The consequence is that we don't have to worry about the (potentially infinite dimensional) feature space — only about the kernel (similarity measure in the input space). Applying this trick yields the following predictive equations:

$$\boldsymbol{\mu}_{y_*|\mathbf{y}} = K(\mathbf{x}_*, \mathbf{X})^\top [K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1} \mathbf{y}, \tag{2.78}$$

$$\boldsymbol{\Sigma}_{y_*|\mathbf{y}} = K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, \mathbf{X})^\top [K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1} K(\mathbf{X}, \mathbf{x}_*). \tag{2.79}$$

The GP predictive distribution is thus given by

$$\begin{aligned} p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(&K(\mathbf{x}_*, \mathbf{X})^\top [K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1} \mathbf{y}, \\ &K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, \mathbf{X})^\top [K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1} K(\mathbf{X}, \mathbf{x}_*)). \end{aligned} \tag{2.80}$$

2.8 Bayesian Optimization and Active Learning

Gaussian Processes are not only powerful models at regression and prediction tasks. Due their probabilistic nature of being able to provide predictions with uncertainty estimates they can also be used in Bayesian Optimization (BO) [148, 149]. BO is a iterative gradient-free optimization method with the goal of estimating the global minimum of some black-box function $f(\boldsymbol{\Theta})$ for $\boldsymbol{\Theta} \in \mathcal{T}$ where $\mathcal{T}$ can be a hybrid space [150]. "Hybrid space" refers to spaces where some dimensions can be discrete (such as batch sizes of neural networks) and other dimensions might be continuous (such as learning rate of

neural networks). Formally, the aim of BO is to find a solution $\mathbf{\Theta}^*$ such that

$$\mathbf{\Theta}^* = \text{argmin}\, f(\mathbf{\Theta}). \tag{2.81}$$

Since our objective is to find the global minimum of the function, $f(\mathbf{\Theta})$ is sometimes referred to as the *objective* function. For BO purposes, it is often the case that function f has no analytical form (e.g. defined implicitly through simulation), hence the name "black box" function. Consequently, we cannot write an equation $f(\mathbf{\Theta}) = ...$ and find its gradients. We can only evaluate the function at that specific point $\mathbf{\Theta}$ and the evaluation itself might be time consuming. An often used specific example of such problem is $\mathbf{\Theta}$ being hyperparameters of a neural network, and $f(\mathbf{\Theta})$ is some validation loss / performance metric after training the network with hyperparameters $\mathbf{\Theta}$. Given large networks and thus long training times (and perhaps monetary restrictions for access to GPU), optimizing for the best set of hyperparameters $\mathbf{\Theta}$ often only has allocated strict limited amount of trials. The problem of hyperparameter tuning can be done with random or grid-search like strategies, where BO aims at finding better hyperparameters on the same or smaller time budget, and indeed, it has been a successful story [151, 152].

Deploying BO requires two crucial decisions from the experimenter: 1) choice of surrogate function and 2) choice of acquisition function. The surrogate function μ can be seen as a "model function" that aims at mimicking the objective function, i.e., $\mu(\mathbf{\Theta}) \approx f(\mathbf{\Theta})$. But the surrogate function in BO also goes beyond predicting the underlying objective by incorporating an uncertainty estimate $\sigma(\mathbf{\Theta})$ for the prediction $\mu(\mathbf{\Theta})$. Introduced in Section 2.7, the Gaussian Process (GP) is a popular choice of surrogate function for BO protocols as [148], but any model capable of providing a distribution over output values meets the requirements of BO including Random Forests (RFs) [153], Deep Ensembles (DEs) [154] or mean-field Bayesian Neural Networks (BNNs) [155]. Although, the GP guarantees the predictive distribution to be a normal distribution, this is not the case for general surrogates. However, a predictive normal distribution is often used where the mean and standard deviation is computed empirically from samples, and we shall assume the same from now on, that is, regardless of surrogate choice we assume

$$p(y|\mathbf{\Theta}_*, \mathcal{D}) = \mathcal{N}(\mu_{\mathcal{D}}(\mathbf{\Theta}_*), \sigma^2_{\mathcal{D}}(\mathbf{\Theta}_*)), \tag{2.82}$$

where $\mathcal{D} = \{\mathbf{\Theta}, f(\mathbf{\Theta})\}_{i=1}^n$ is a dataset of observations and $\mu_{\mathcal{D}}$ and $\sigma_{\mathcal{D}}$ refer to a mean and standard deviation function trained on $\mathcal{D}$. Furthermore, we model the output as potentially being noisy observations of the true underlying objective function $y = f(\mathbf{\Theta}) + \epsilon$ for additive noise $\epsilon \sim \mathcal{N}(0, \sigma^2_{noise})$.

The mean $\mu(\boldsymbol{\Theta})$ and uncertainty $\sigma(\boldsymbol{\Theta})$ from the surrogate function are key ingredients in how the BO procedure sequentially chooses which hyperparameters $\boldsymbol{\Theta}$ to try out via the acquisition function. An acquisition function is thus a function which takes the entire history $\mathcal{D}$ as input, and returns a function over $\boldsymbol{\Theta}$ which is then used to choose the $\boldsymbol{\Theta}_{next}$ that maximizes the acquisition function. A popular choice of acquisition function include the expected improvement (EI) which has the form

$$\text{EI}(\boldsymbol{\Theta}) = (\mu(\boldsymbol{\Theta}) - f(\boldsymbol{\Theta}^+))\Phi(Z(\boldsymbol{\Theta})) + \sigma(\boldsymbol{\Theta})\phi(Z(\boldsymbol{\Theta})), \tag{2.83}$$

where $Z(\boldsymbol{\Theta}) = \frac{\mu(\boldsymbol{\Theta})-f(\boldsymbol{\Theta}^+)}{\sigma(\boldsymbol{\Theta})}$, $f(\boldsymbol{\Theta}^+)$ denotes the minimum observed objective value, Φ is the cumulative distribution function (CDF) a standard normal distribution and ϕ is the probability density function (PDF) of a standard normal distribution. The EI in Eq. (2.83) is motivated by finding the $\boldsymbol{\Theta}$ which on average maximizes the improvement $I(\boldsymbol{\Theta}) = \max(f(\boldsymbol{\Theta}^+) - \mu(\boldsymbol{\Theta}), 0)$. Plugging in the GP predictive distribution into $\mathbb{E}[\max(f(\boldsymbol{\Theta}^+) - \mu(\boldsymbol{\Theta}), 0)]$ yields Eq. (2.83) (see [156] for proof). Given a history of observations $\mathcal{D}$, the BO procedure works as follows. Choose a surrogate function with predictive distribution $p(y|\boldsymbol{\Theta}_*, \mathcal{D}) = \mathcal{N}(\mu(\boldsymbol{\Theta}), \sigma^2(\boldsymbol{\Theta}))$ and acquisition function $\text{Acq}(\boldsymbol{\Theta})$. Iteratively repeat the following steps until convergence or time/monetary budget has been met:

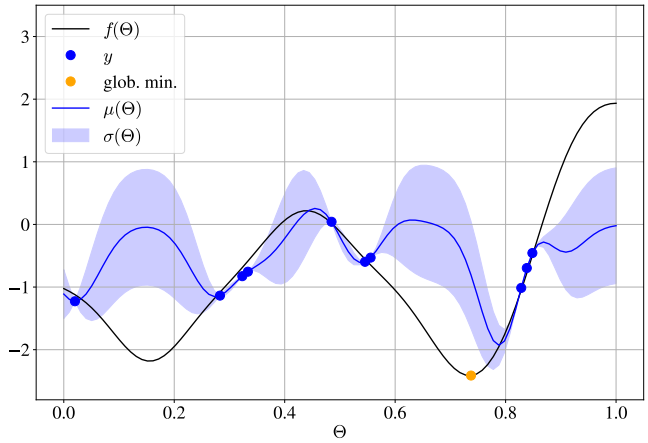
(i) Fit surrogate function to data $\mathcal{D}$

(ii) Choose next input data point according to $\boldsymbol{\Theta}_{next} = \text{argmax}_{\boldsymbol{\Theta}} \text{Acq}(\boldsymbol{\Theta})$ for example using expected improvement: $\text{Acq}(\boldsymbol{\Theta}) = \text{EI}(\boldsymbol{\Theta})$

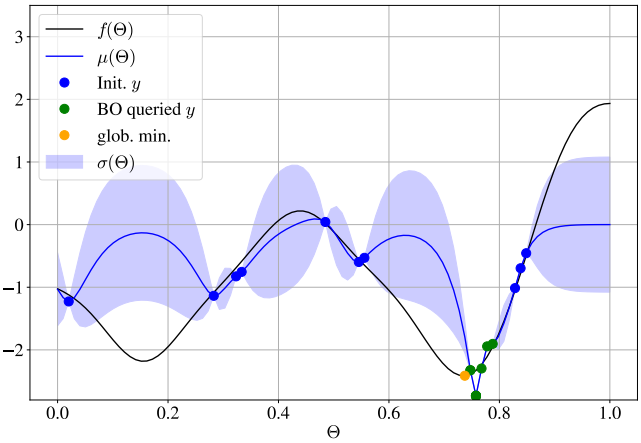
(iii) Evaluate the objective function at $f(\boldsymbol{\Theta}_{next})$

(iv) Update data history $\mathcal{D}$ with $(\boldsymbol{\Theta}_{next}, f(\boldsymbol{\Theta}_{next})$

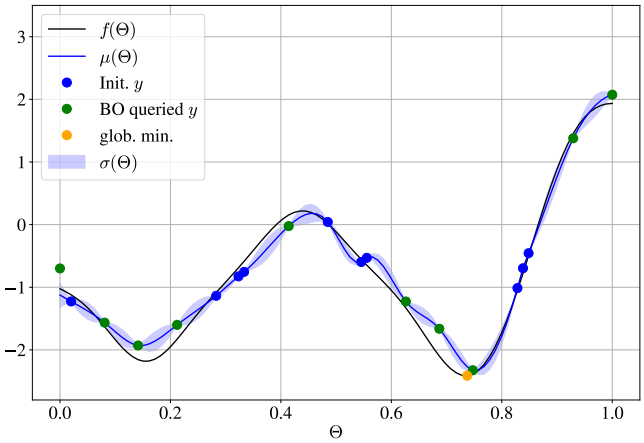
Active Learning Active learning (AL) [157] is very similar to Bayesian Optimization (BO), except for the acquisition functions being slightly different. Whereas the goal of BO is to find the global minimum/maximum of a function $f(\boldsymbol{\Theta})$, the goal of AL is to end up with a predictive distribution $p(y|\boldsymbol{\Theta}_*, \mathcal{D})$ that approximates the underlying objective function. Thus AL is also sometimes called *optimal experimental design*, since it is an active and iterative way of choosing/querying as few samples as possible from a function in order to learn as much as possible [157]. One acquisition strategy is uncertainty sampling, which queries the point according to $\text{argmax}_{\boldsymbol{\Theta}}\, \sigma(\boldsymbol{\Theta})$. A comparison between how samples are queried for BO and AL is provided in Fig. 2.6.



(a) Initial model fitted on objective $y(\Theta) = f(\Theta) + \epsilon$ with noisy nature.



(b) Model and samples after BO queries



(c) Model and samples after AL queries

Figure 2.6: Comparison between samples and models from BO with expected improvement acquisition **(b)** and AL with uncertainty sampling **(c)**. Comparing to BO, AL aims at getting a good fitting of the underlying function and thus samples more uniformly to minimize surrogate uncertainty whereas BO aims at finding the global minimum.

2.9 Unsupervised Learning with Restricted Boltzmann Machine

In Sections 2.6-2.8 we dealt with supervised machine learning scenarios which is an important part of the machine learning success history and is relevant for papers C and D. But there also exists a class of models and approaches, equally (and sometimes argued even more) important, known as unsupervised learning. Here we have datapoints $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]$ but with no class labels or targets, i.e., no "supervision". Unsupervised learning is important for more general pattern recognition and generation [158]. One type of generative model known as the *restricted Boltzmann machine* (RBM) is of particular interest for NISQ computing as this is closely related to the statistical mechanics discussed in Sections

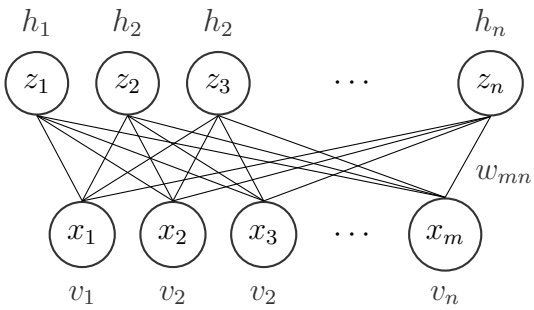


Figure 2.7: Depiction of Restricted Boltzmann Machine (RBM). Adapted from [160].

2.3-2.5. RBMs are generative by means of their ability to learning patterns in data and being able to generative new ones with same characteristics. More specifically, from samples $\mathbf{x} \sim p(\mathbf{x})$ a RBM aims at inferring a parameterized model $p_{\boldsymbol{\theta}}(\mathbf{x}) \approx p(\mathbf{x})$ such that new samples $\mathbf{x}_*$ generated from $p_{\boldsymbol{\theta}}$ follows $\mathbf{x}_* \sim p(\mathbf{x})$. It does so by employing a neural network model consisting of *visible* neurons $\mathbf{x} = \{x_1, x_2, ..., x_V\}$ and *hidden* neurons $\boldsymbol{z} = \{z_1, z_2, ..., z_H\}$. The "restrictiveness" of RBM comes from the fact that there only exists connections between visible and hidden neurons, and thus no connections in between the visibles as is the case for general Boltzmann machines . Normally, neurons in the RBM are binary variables, but in principle this can also be relaxed to continuous values [159]. We consider only binary variables for now. The energy function of the RBM is given in Eq. (2.84) which is equivalent to the one in Eq. (3.7).

$$\begin{aligned} E(\mathbf{x}_v, \boldsymbol{z}_h) &= -\sum_h b_h z_h - \sum_v b_v x_v - \sum_h \sum_v w_{vh} z_h x_v \\ &= -\mathbf{b}_h^\top \boldsymbol{z} - \mathbf{b}_v^\top \mathbf{x} - \mathbf{x}^\top \mathbf{W} \boldsymbol{z} \end{aligned} \tag{2.84}$$

where the RBM parameters $\boldsymbol{\theta} = \{\mathbf{b}_h, \mathbf{b}_v, \mathbf{W}\}$ are learned from a specific dataset. The probability of observing a specific combination $(\mathbf{x}_v, \boldsymbol{z}_h)$ is given by

$$p(\mathbf{x}, \boldsymbol{z}) = \frac{1}{\sum_{\mathbf{x}', \boldsymbol{z}'} e^{-E(\mathbf{x}', \boldsymbol{z}')}} e^{-E(\mathbf{x}, \boldsymbol{z})} \tag{2.85}$$

which is exactly the Boltzmann distribution in Eq. (2.55) for partition function $Z = \sum_{\mathbf{x},\boldsymbol{z}} e^{-E(\mathbf{x},\boldsymbol{z})}$. We can now obtain the marginal distribution over the visible units marginalizing over the latent units

$$p(\mathbf{x}) = \frac{1}{Z} \sum_{\boldsymbol{z}} e^{-E(\mathbf{x},\boldsymbol{z})} \tag{2.86}$$

Taking the log on both sides yields the log likelihood

$$\begin{aligned} \log p(\mathbf{x}) &= \log \frac{1}{Z} \sum_{\boldsymbol{z}} e^{-E(\mathbf{x},\boldsymbol{z})} \\ &= \log \frac{1}{Z} + \log \sum_{\boldsymbol{z}} e^{-E(\mathbf{x},\boldsymbol{z})} \\ &= -\log Z + \log \sum_{\boldsymbol{z}} e^{-E(\mathbf{x},\boldsymbol{z})} \\ &= \underbrace{-\log \sum_{\mathbf{x},\boldsymbol{z}} e^{-E(\mathbf{x},\boldsymbol{z})}}_{\text{Unclamped } \mathcal{F}_u} + \underbrace{\log \sum_{\boldsymbol{z}} e^{-E(\mathbf{x},\boldsymbol{z})}}_{\text{Clamped } \mathcal{F}_c} \end{aligned} \tag{2.87}$$

that is, log likelihood equals the equilibrium (unclamped) free energy minus the free energy when the visible units are clamped to the datapoint $\mathbf{x}$; both at temperature $\beta = 1$. Derivatives of these energies turns out to be expectations over inner products, that is, correlations between visible and hidden units. The gradient of the unclamped free energy w.r.t. one of the parameters is the expected value of $\boldsymbol{z}^\top \mathbf{x}$,

$$\nabla_\theta \mathcal{F}_c = \mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z}|\mathbf{x})}[\boldsymbol{z}^\top \mathbf{x}] \tag{2.88}$$

which is easy to do. However for the unclamped part,

$$\nabla_\theta \mathcal{F}_u = \mathbb{E}_{\boldsymbol{z},\mathbf{x} \sim p(\boldsymbol{z},\mathbf{x})}[\boldsymbol{z}^\top \mathbf{x}], \tag{2.89}$$

it is generally intractable and has to be estimated [161].

We close this chapter by summarizing that these sections constituted the main background theory necessary to follow the next chapters in this thesis. We now move on to elaborate on the main topic, parameterized quantum circuits, before we explain the scientific contributions.

Chapter 3

Quantum Neural Networks

WITH our established notation outlined in the previous chapter, we are ready to embark on the main focus of this Ph.D. project: quantum neural networks (QNNs). A slightly less buzzing name for these types of models are parameterized quantum circuits (PQCs) or variational quantum algorithms (VQAs) but we shall use all three terms interchangeably, just as done extensively the literature [162, 163, 164, 165], albeit there are key differences between general VQAs and classical neural networks as we shall see.

We saw in Chapter 2 that a quantum computers takes an initial state, represented as a density matrix $\rho_{init} \in \mathbb{C}^{2^N \times 2^N}$ for N qubits, and produces an output state of same dimension $\rho_{out} \in \mathbb{C}^{2^N \times 2^N}$ via a series of gate operations acting on the initial state. The output state is measured in the computational state (each spin yields up or down) which collapses the state onto one of the computational basis states (Postulate 4). In order to create a quantum algorithm with specific application, careful consideration needs to go in to what gates are applied in what order to obtain the optimal cocktail of superposition and interference such that the resulting entangled output state is useful. Often, however, the result is a (very) deep circuit which is not preferable for the noisy quantum hardware with few qubits we have access to now and in the near-future.

There is, however, another way to go about creating quantum algorithms; an idea which also fits well into the idea of creating *shallow* circuits being more resilient to noise. Recall that in Chapter 2 we learned that the gates can be represented with a unitary matrix σ, which can be parameterized with parameter θ_i, such that $U(\theta_i) = e^{i\theta_i\sigma}$. Acting with the gate $U(\theta_i)$ to the state ρ, corresponds to changing the density matrix according to $\rho' = U(\theta_i)\rho U(\theta_i)^\dagger$. Without loss of generality, we can draw a general unitary circuit as done in Fig. 3.1. Despite we in practice most often use only one- and two qubit gates

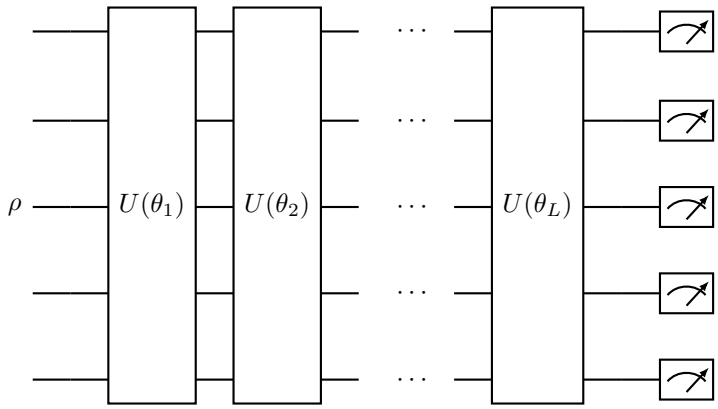


Figure 3.1: General parameterized unitary circuit with parameters $\boldsymbol{\theta} = \{\theta_1, \theta_2, ..., \theta_L\}$.

due to the aforementioned Solovay-Kitaev theorem [83], we often depict general unitary gates U as acting on all the qubits since this simply corresponds to $U = \mathbb{1} \otimes ...U_i... \otimes \mathbb{1}$ where U_i is a local gate acting on one or two qubits and the rest are left alone revealed by the identity acting on the remaining qubits. Mathematically, we represent the *overall* circuit unitary as

$$U(\boldsymbol{\theta}) = U(\theta_L)U(\theta_{L-1})...U(\theta_2)U(\theta_1) \tag{3.1}$$

We often write the state after all unitary gates but prior to measurement as $\rho(\boldsymbol{\theta})$, and in many of the first VQA papers it was assumed that the initial state ρ (see Fig. 3.1) was a pure state $|\psi_{init}\rangle = \boldsymbol{\psi}_{init}$ as well as—due to the short depth—the quantum circuit could keep unitary, creating the state

$$|\psi(\boldsymbol{\theta})\rangle = U(\theta_L)U(\theta_{L-1})...U(\theta_2)U(\theta_1)\,|\psi_{init}\rangle \tag{3.2a}$$

$$\boldsymbol{\psi}(\boldsymbol{\theta}) = U(\theta_L)U(\theta_{L-1})...U(\theta_2)U(\theta_1)\boldsymbol{\psi}_{init} \tag{3.2b}$$

where $|\psi_{init}\rangle = \boldsymbol{\psi}_{init}$ often is the computational basis zero-state for N qubits denoted $|0\rangle^{\otimes N} = \boldsymbol{e}_0^{\otimes N}$ (see Eq. (2.3)). Note that Eq. (3.2) corresponds to the overall circuit unitary in Eq. (3.1) acting on the initial state. The left hand side in Eq. (3.2) is sometimes referred to as an *ansatz* state, which is another name for a *trial state* since a specific $\boldsymbol{\theta}$ creates an initial "trial" out of many states. $U(\boldsymbol{\theta})$ is thus called the *ansatz* and we explore ideas on how to design these in Section 3.3.1. The goal of VQAs is to use a classical computer to optimize $\boldsymbol{\theta}$ in Eq. (3.1), and this is what in 2014 sparked a new field of quantum machine learning (QML) referred in this thesis to as second-wave QML, or more generally second-wave

algorithms.

Although, Google in 2019 demonstrated quantum supremacy [166] for a proof-of-concept task, we still do not have a seen experimental evidence for which second-wave quantum algorithms running on noisy intermediate-scale quantum (NISQ) computers are superior. VQAs are generally considered the best candidates for achieving computational advantages in the near-term for practical problems [167]. The research moves very fast and almost weekly we see new bold [168], concerning [169] and very encouraging [170, 171] results and ideas. This roller coaster nature may indicate there are still a lot of unknowns areas of QML waiting to be explored. In the remainder of this chapter, we dive into the technical details of PQCs/QNNs/VQAs and highlight some of the key results many of which are provided to give an overview as well as outline the work that sparked the ideas for the scientific contributions of this thesis (Chapter 4-Chapter 7).

3.1 Hybrid Quantum-Classical Computation

As mentioned in the last part of Chapter 1, in 2014, two independent papers were published and together they shifted the field of QML from designing fully error-corrected quantum circuit (i.e. first-wave QML) into designing shallow circuits that solve problems by partnering up with a classical computer: Peruzzo et al. [108] introducing the variational quantum eigensolver (VQE) and Farhi et al. [107] introducing the quantum approximate optimization algorithm (QAOA). Both papers were first instances of a more general group of algorithms (PQCs/VQAs/QNNs), which are *hybrid quantum-classical computational* protocols. The idea of PQCs is to have the quantum processor performing short bursts of *parameterized* computation, measure the qubits and then based on those measurement statistics, update the parameters inside the circuits such that some loss function is minimized (see Fig. 3.2). Limiting the protocol to perform shallow computations means that fewer errors can accumulate in the output state. The loss function encodes our problem of interest such that when we minimize an expectation we get closer to a good solution. The most commonly used loss function is the expectation value of some energy operator $\mathcal{H}$. The VQE [108] uses a Hamiltonian of a molecule to find the groundstate of that molecule, which is a hard task in quantum chemistry. The QAOA [107] encodes the combinatorial problem—the so-called MaxCut problem [172]—in a Hamiltonian such that the solution encodes how one would cut a graph with maximum distance between the two sets of nodes. It was later shown that these types of circuits (VQAs) are universal quantum computing models [173, 174], and QAOA was later showed

to—even with low depth circuits—produce probability distributions not efficiently simulated on classical devices [175]. Furthermore, there has been interesting results from portfolio optimization in finance applications [176], and, as we shall see momentarily, quantum machine learning [177]. Another advantage of VQAs is that they are inherently resilient to coherent/systematic noise c in the gates $U(\theta + c)$ since the classical optimizer should be able to learn how to correct this when minimizing the loss such that it converges to $\theta^* = \theta - c$ [178, 179].

The problem of finding minimum eigenvalues of large matrices is a well-known problem [180] with applications in many fields, including e.g. machine learning and quantum chemistry, and it is generally argued that an important application of quantum computers is to find groundstates of Hamiltonians [181]. PQCs aim at this task by using a classical optimizer to learn the quantum circuit that prepares the groundstate. PQCs are thus often called QNNs due to their input-to-output structure, their (unitary) matrix multiplication and their training via stochastic gradient descent. However, one crucial ingredient—which is what gives classical neural network most of their power—is missing from general VQAs: non-linearities. The idea of QNNs is not new [182] and people have been searching for meaningful quantum counterparts to classical neural works for a while [183, 184, 185]. The results include ideas such as feed forward networks [186] and quantum convolutional neural networks (QCNNs) [187] where operations such as measurements and conditional unitary operations create non-linearities [188]. These ideas can be more generally applied in quantum computing to create non-linearities [189] and Wan et al. showed that classical neural networks can be embedded into QNNs [190]. Furthermore, despite gates are unitary and the state consequently evolves linearly, the expectation value of some observables can be changed in a non-linear way [191].

Another way to interpret some PQCs are as kernel methods when trained supervised [192]. Since our input state needs to encode the data points, we are essentially embedding data into a high dimensional (2^N) vector space using only N qubits. What kind of vector space? Since wave vectors and density matrices are complex (see Postulate 1) and there exists an inner product (for example giving us the normalization, see Eq. (2.5)), state vectors and density matrices live in a *Hilbert* space. If one embeds a datapoint $\mathbf{x}_i \in \mathbb{C}^{2^N}$ as the initial quantum state at the beginning of our quantum circuit, and subsequently updates the circuit $U(\boldsymbol{\theta})$ such that the expectation value of one of the qubits reflects the corresponding classification label y_i, then the QNN essentially computes inner products and is thus indeed a kernel method [193]. In fact, there is a whole line of literature trying to use a quantum computer as a kernel (i.e. computing inner products) and use this kernel in predictive tasks [194]. Other research considers

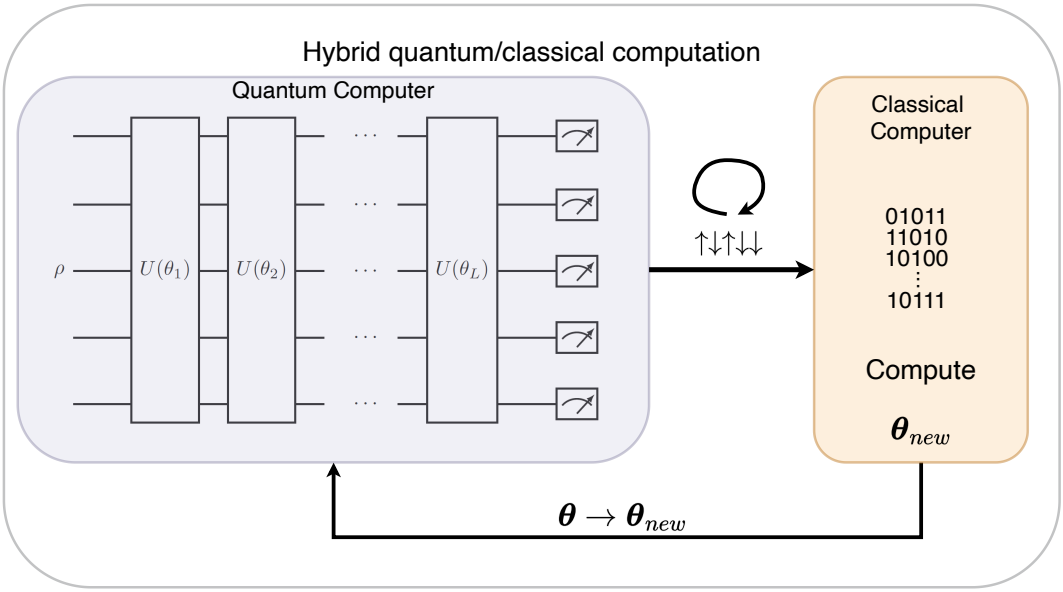


Figure 3.2: Hybrid quantum/classical computation. An initial quantum state ρ (often all qubits are put in the zero / spin up state) is passed through a paramterized quantum circuit, the spins are measured and our measurement apparatus reads the eigenvalues of the Pauli Z matrix(spin up = 1, spin down = -1), this process is repeated to gain measurement statistics which is then used to update the circuit parameters such that a predefined loss function is minimized.

using quantum circuits as general machine learning models [195] or generative models [196], and in fact some generative tasks have been implemented on actual hardware [197].

The key difference between first-wave quantum algorithms and second-wave algorithms (see Chapter 1) is that second-wave algorithms designs are not just looking at one particular circuit but in fact a family of circuits $U(\boldsymbol{\theta})$ which needs to be optimized in order to do some task, similar to how classical neural networks operate. This family of circuits is sometimes also called a *hypothesis class* since we hypothesize that a solution the problem lies in all the functions that $U(\boldsymbol{\theta})$ span. Whether the solution actual lies inside $U(\boldsymbol{\theta})$ is not given and not at all trivial to express for general Hamiltonians, but we return to this in Section 3.3.1. We summarize this section with Fig. 3.2 illustrating the general idea of quantum-classical hybrid computation, where we use the following set of steps that are repeated until convergence:

(i) Quantum computer makes one shot (a forward-pass) to prepare spins in the state $\rho(\boldsymbol{\theta})$

(ii) Spins are measured (often in the computational basis)

(iii) Step (i) and (ii) is repeated to obtain measurement statistics

(iv) Use a loss function together with measurement statistics to update circuit parameters $\boldsymbol{\theta} \rightarrow \boldsymbol{\theta}_{new}$

In the next section, we look at some of the most used loss functions.

3.2 Loss Function

The most commonly used loss function in VQAs is some energy operator $\mathcal{H}$ that encodes the problem at hand. For example it could be the Ising model (Eq. (3.7)), and hence the goal of optimizing $\boldsymbol{\theta}$ is to find the groundstate of $\mathcal{H}$. The general framework is that we have a state over N qubits is prepared by a parameterized circuit $|\psi(\boldsymbol{\theta})\rangle = \boldsymbol{\psi}(\boldsymbol{\theta})$ and a corresponding loss function being an energy operator $\mathcal{H}(\boldsymbol{C})$ is given by its Pauli decomposition

$$\mathcal{H}(\boldsymbol{C}) = \sum_{m=1}^{M} c_m P_m, \tag{3.3}$$

with coefficients $\boldsymbol{C} = \{c_1, c_2, ..., c_M\}$ and where P_m is an observable (Hermitian matrix) that can be written as a tensor product of exactly N one-qubit Pauli matrices (Eq. (2.22)) acting on at most k qubits. We call these *local* Hamiltonians: k-local Hamiltonians means operators decomposable into local operators acting on k spins at max, and with the identity on the rest. We will almost exclusively be considering 2-local Hamiltonians, such as the one written in Eq. (2.48), which can be written even more compactly as

$$\mathcal{H}(\boldsymbol{C}) = \sum_{(\alpha_i,\alpha_j)} \sum_i J_{ij}^{(\alpha_i,\alpha_j)} \sigma_i^{\alpha_i} \sigma_j^{\alpha_j} + \sum_{\alpha_i} \sum_i b_i^{\alpha_i} \sigma_i^{\alpha_i} \tag{3.4}$$

for $\boldsymbol{C} = \{J_{ij}^{(\alpha_i,\alpha_j)}, b_i^{\alpha_i}\}$ and $\alpha_i \in \{X, Y, Z\}$ corresponds to a Pauli measurement (Eq. (2.22)) on the i'th qubit, for example, σ_4^X corresponds to a spin left/right measurement on qubit 4. The expectation $\langle \mathcal{H}(\boldsymbol{C})\rangle$ is found by finding the expectation of the individual Pauli terms in the Hamiltonian,

$$\langle \mathcal{H}(\boldsymbol{C})\rangle := \langle \psi(\boldsymbol{\theta})|\mathcal{H}(\boldsymbol{C})|\psi(\boldsymbol{\theta})\rangle = \sum_{(\alpha_i,\alpha_j)} \sum_i J_{ij}^{(\alpha_i,\alpha_j)} \langle \psi(\boldsymbol{\theta})|\sigma_i^{\alpha_i}\sigma_j^{\alpha_j}|\psi(\boldsymbol{\theta})\rangle + \sum_{\alpha_i} \sum_i b_i^{\alpha_i} \langle \psi(\boldsymbol{\theta})|\sigma_i^{\alpha_i}|\psi(\boldsymbol{\theta})\rangle, \tag{3.5a}$$

$$\langle \mathcal{H}(\boldsymbol{C})\rangle := \boldsymbol{\psi}(\boldsymbol{\theta})^H \mathcal{H}(\boldsymbol{C}) \boldsymbol{\psi}(\boldsymbol{\theta}) = \sum_{(\alpha_i,\alpha_j)} \sum_i J_{ij}^{(\alpha_i,\alpha_j)} \boldsymbol{\psi}(\boldsymbol{\theta})^H \sigma_i^{\alpha_i}\sigma_j^{\alpha_j} \boldsymbol{\psi}(\boldsymbol{\theta}) + \sum_{\alpha_i} \sum_i b_i^{\alpha_i} \boldsymbol{\psi}(\boldsymbol{\theta})^H \sigma_i^{\alpha_i} \boldsymbol{\psi}(\boldsymbol{\theta}). \tag{3.5b}$$

Despite interesting methods have been proposed to minimize the number of samples required [198], it generally takes a lot of shots/forward-passes/repetitions of the quantum circuit with the same parameters $\boldsymbol{\theta}$ in order to get stable statistics necessary for some applications. We return to this issue in Section 3.4, but the number of samples is typically large because each term/sub-expectation in the above sum needs needs an amount scaling with $\frac{1}{\epsilon^2}$ for error ϵ and given that we change the state upon measurement (see Postulate 4) we—worst case—need to run the circuit again for another term/sub-expectation.

Finding the circuit parameters $\boldsymbol{\theta}^*$ that minimizes Eq. (3.3) corresponds to the optimization problem

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \langle \psi(\boldsymbol{\theta}) | \mathcal{H}(\boldsymbol{C}) | \psi(\boldsymbol{\theta}) \rangle , \tag{3.6a}$$

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \boldsymbol{\psi}(\boldsymbol{\theta})^H \mathcal{H}(\boldsymbol{C}) \boldsymbol{\psi}(\boldsymbol{\theta}). \tag{3.6b}$$

If the groundstate is contained inside $U(\boldsymbol{\theta})$, obtaining $\boldsymbol{\theta}^*$ from the above equation yields our circuit unitary $U(\boldsymbol{\theta}^*)$ to prepare the groundstate (i.e. the minimum energy eigenstate) of $\mathcal{H}(\boldsymbol{C})$ and when we compute Eq. (3.3) we get the minimum eigenvalue of $\mathcal{H}(\boldsymbol{C})$. There are also approaches to compute higher energy eigenvalues (excited states) such as first estimating the groundstate and subsequently changing the loss function to contain a penalty on the groundstate which yields slightly different parameter update rules for the classical optimizer [199] or using hierarchical approaches [200]. Since finding eigenvectors of matrices is at the very core of some machine learning models such as Principal Component Analysis (PCA), there has been interesting ideas on how to diagonalize density matrices and to obtain both eigenvalue and eigenvectors [201, 202] as well as how to actually implement the covariance matrix of the dataset [203]. Furthermore, as mentioned in Chapter 1, we saw the HHL algorithm [91] was able to solve linear systems of equations using deep fault-tolerant quantum algorithms, but it turns out that this is also possible to reformulating it as a energy minimization problem [204]. In Section 3.3 we elaborate on techniques to estimate $\boldsymbol{\theta}^*$.

Up until VQE was introduced, Quantum Phase Estimation (QPE) [205, 206] was the leading algorithm in estimating eigenvalues of Hermitian matrices using quantum computers having an exponential speedup compared to classical exact diagonalization [207]. However, since QPE requires deep circuits it is not appropriate for NISQ devices [208]. Instead, given that many real-world quantum systems have a sparse Hamiltonian, the expectation (Eq. (3.5)) often has at most a polynomial number of Pauli terms relative to the number of spins making them effective to estimate on a quantum computer [167]. From

the adiabatic theorem in Section 2.3.1, we learned that the speed at which we can walk on the analog adiabatic pathway is inversely proportionally to the spectral gap in the Hamiltonian, and indeed, the complexity of solving the general VQE problem is related to this property of the Hamiltonian. If there is a spectral gap large enough (inverse-polynomial in the number of spins) in our Hamiltonian, finding its groundstate using VQAs is QMA-complete for $k \geq 2$ [209]. Quantum Merlin-Arthur (QMA) is the quantum-extention of the probabilistic nondeterministic polynomial (NP) class—called MA— which contain problems hard for even quantum computers to solve but given some constraints a solution can be easily verified with a quantum computer [210]. We return to the practical difficulties for VQA in Section 3.3 but note already now that no obvious existence of exponential speedups with VQAs is at hand.

Essentially what VQAs are trying to accomplish is to use a quantum computer to *simulate* a quantum system. This task was shown possible to do for a quantum computer when the quantum system of interest are spins [211]. Although Nature "wants" systems to be in the lowest energy state, it is often a hard task to perform in reality: even for rather simple Hamiltonians the system is likely to get stuck in local minima of the *energy landscape*. For machine learners, this is not a new phenomenon as it is observed all the time in computer science optimization problems, for example optimizing neural networks [212]. If a local minimum has an energy very close to the global minimum, it might be an acceptable solution for some applications, however, problems might arise in some applications when the loss function has many local optima significantly different from the global minimum.

In the following we introduce some often used loss functions for QNNs, which is also used in the Paper A-C.

Ising Spin Ring The Ising Spin Ring model is given by $\mathcal{H}_{\text{TFI}} = \mathcal{H}_M + \mathcal{H}_C$ for

$$\mathcal{H}_M = -\sum_i h_i X_i, \quad \mathcal{H}_C = -\sum_i J_i Z_i Z_{i+1}. \tag{3.7}$$

such that the spins are assumed to be on a lattice ring where it is sufficient to only model nearest neighbor interactions.

Transverse-field Ising Spin Ring The transverse-field Ising (TFI) Spin Ring model is given by $\mathcal{H}_{\text{TFI}} = \mathcal{H}_M + \mathcal{H}_C$ for

$$\mathcal{H}_M = -\sum_i h_i X_i, \quad \mathcal{H}_C = -\sum_i J_i Z_i Z_{i+1}. \tag{3.8}$$

Heisenberg XXZ Spin Ring The Heisenberg XXZ spin ring model is given by $\mathcal{H}_{\text{XXZ}} = \mathcal{H}_M + \mathcal{H}_C$ for

$$\mathcal{H}_M = \sum_{i=1}^{N} h_i Z_i, \; \mathcal{H}_C = \sum_{i=1}^{N} [X_i X_{i+1} + Y_i Y_{i+1} + Z_i Z_{i+1}].$$

Lithium Hydride The Lithium Hydride (LiH) Hamiltonian has a linear combination of 6-local Pauli strings $P_k \in \{\mathbb{1}, X, Y, Z\}$ as

$$\mathcal{H}_{LiH} = \sum_{k=1}^{r_h} h_k P_k. \tag{3.9}$$

3.2.1 Free Energy as Loss Function

Another choice of loss function for VQAs is the free energy [163] which we derived in Eq. (2.59). There are very deep connections to statistical learning theory, information theory, Bayesian inference and more beyond the scope of this thesis but the curious reader could be directed to the lecture notes by Jose et al. [213] or how the physics of energy based models work by Huembeli et al. [214]. Minimizing the free energy, rather than the energy, yields the qubits to approximate the thermal state (Eq. (2.62)) rather than the groundstate [215]. This is because the relative entropy D (also known as the Kullback-Liebler divergence) between the circuit created state $\rho(\boldsymbol{\theta})$ and the Gibbs state at inv. temperature β denoted ρ_β can be written as the difference between their free energies $F_{\boldsymbol{\theta}}$ and F_β

$$D(\rho(\boldsymbol{\theta})||\rho_\beta) = \text{Tr}[\rho(\boldsymbol{\theta}) \log \rho(\boldsymbol{\theta})] - \text{Tr}[\rho(\boldsymbol{\theta}) \log \rho_\beta] = \beta(F_{\boldsymbol{\theta}} - F_\beta) \geq 0 \tag{3.10}$$

which has its minimum ($D = 0$) when $\rho(\boldsymbol{\theta}) = \rho_\beta$. We shall use the term *variational quantum thermalizers* (VQTs) to denote QNNs that aims at preparing thermal states of energy operators at some inverse temperature β. In fact, the very first contribution (Paper A, Chapter 4) is centered around using the free energy as loss function to approximate a thermal state of various spin models. Although the field of groundstate preparation is rich and tons of other popular ideas such as qubitization [216] exist, we limit our focus to VQE and VQT.

There is a caveat when minimizing the free energy rather than the energy: the entropy is not an

observable and thus has to be estimated. Recall from Postulate 4 that observable operators needs to be Hermitian, and the entropy (Eq. (2.51)) is not. Paper A (Chapter 4) lists ways to estimate the entropy and thus use the free energy as loss function in practice.

3.3 Optimization

How QNNs are optimized has at least two interpretations: a "machine learning" and a "physical" approach. The update rules turns out to be the same, and it is only of matter of *how* we get to the various update rules proposed [217].

The machine learner can think of finding an optimal $\boldsymbol{\theta}$ as an optimization task where we can use gradient based methods, just as we do in deep learning. That is, for infinitesimal changes in a single circuit θ_i, this expectation changes as

$$\partial\theta_i := \frac{\partial}{\partial\theta_i} \langle\psi(\boldsymbol{\theta})| \mathcal{H}(\boldsymbol{C}) |\psi(\boldsymbol{\theta})\rangle \tag{3.11a}$$

$$\partial\theta_i := \frac{\partial}{\partial\theta_i} \boldsymbol{\psi}(\boldsymbol{\theta})^H \mathcal{H}(\boldsymbol{C}) \boldsymbol{\psi}(\boldsymbol{\theta}) \tag{3.11b}$$

Schuld et al. [218] showed that these gradients can be obtained analytically from the quantum computer using the *parameter-shift rule*,

$$\begin{aligned}\partial\theta_i = &\frac{1}{2}\left\langle\psi\left(\boldsymbol{\theta}+\frac{\pi}{2}\boldsymbol{e}_i\right)\middle|\mathcal{H}(\boldsymbol{C})\middle|\psi\left(\boldsymbol{\theta}+\frac{\pi}{2}\boldsymbol{e}_i\right)\right\rangle \\ &-\frac{1}{2}\left\langle\psi\left(\boldsymbol{\theta}-\frac{\pi}{2}\boldsymbol{e}_i\right)\middle|\mathcal{H}(\boldsymbol{C})\middle|\psi\left(\boldsymbol{\theta}-\frac{\pi}{2}\boldsymbol{e}_i\right)\right\rangle\end{aligned} \tag{3.12}$$

where $\boldsymbol{e}_i$ is an indicator vector (zero everywhere except a one at position i) of length L (number of circuit parameters). In order to minimize the loss/expectation, the parameters are updated in the opposite direction of the gradient, namely using gradient descent the n'th update iteration

$$\boldsymbol{\theta}^{(n+1)} = \boldsymbol{\theta}^{(n)} - \eta\partial\boldsymbol{\theta}^{(n)} \tag{3.13}$$

for some learning rate η. Since expectation values like the ones in Eq. (3.12) always are experimentally evaluated from finite sample sizes, it is argued to be analogous to stochastic gradient descent (SGD) [219] although one could argue it is a slightly different source of stochasticity. Ideas on how to expand

this parameter-shift rule using the geometry of the circuit was also shown to obtain multi-parameter gate gradients on quantum hardware [220]. Although quantum circuits can be complicated and it is hard to guarantee SGD convergence, it has been shown that gradient based method are better for some low-depth circuits when comparing to gradient-free approaches [221].

For quantum systems, we learned that the probability distribution of observing the qubits in the i'th computational basis state was given by the i'th diagonal element of the density matrix, that is,

$$p(i|\boldsymbol{\theta}) = [\text{diag}\rho(\boldsymbol{\theta})]_i. \tag{3.14}$$

The machine learner might use this to come up with higher order optimization methods such as natural gradient descent [222], which optimizes the parameters by transforming the gradients using the *Fisher Information Matrix* $\mathbf{F}$

$$\boldsymbol{\theta}^{(n+1)} = \boldsymbol{\theta}^{(n)} - \eta \mathbf{F}^{-1}(\boldsymbol{\theta}) \partial \boldsymbol{\theta}^{(n)} \tag{3.15}$$

where $\mathbf{F}$ is an approximation to the Hessian [223] of the Kullback-Lieber divergence between $p(i|\boldsymbol{\theta})$ and $p(i|\boldsymbol{\theta} + \boldsymbol{\delta})$ for infitesimal changes to the parameters $\boldsymbol{\delta}$ [224]. The k'th row and l'th column of the Fisher matrix $\mathbf{F}$ is given by

$$[\mathbf{F}(\boldsymbol{\theta})]_{kl} = \sum_i p(i|\boldsymbol{\theta}) \frac{\partial \log p(i|\boldsymbol{\theta})}{\partial \theta_k} \frac{\partial \log p(i|\boldsymbol{\theta})}{\partial \theta_l}, \tag{3.16}$$

which describes how the probability of observing the i'th spin is affected by changing the l'th and k'th parameter. The inverse of Eq. (3.16) adjusts the gradient in order to respect co-dependence between parameters when changing the probability distribution $p(i|\boldsymbol{\theta})$. And exactly because $p(i|\boldsymbol{\theta})$ is isometric to a classical probability distribution, $\mathbf{F}$ is referred to as the *classical* Fisher Information Matrix.

The physicist might think of optimizing the parameters with a different perspective. While evolving a pure quantum state in a closed system for a discrete time step δt yields a unitary matrix multiplied with the state,

$$|\psi(\boldsymbol{\theta}(t + \delta t))\rangle = e^{-i\delta t H} |\psi(\boldsymbol{\theta}(t))\rangle, \tag{3.17a}$$

$$\boldsymbol{\psi}(\boldsymbol{\theta}(t + \delta t)) = e^{-i\delta t \mathbf{H}} \boldsymbol{\psi}(\boldsymbol{\theta}(t)), \tag{3.17b}$$

a continuous state change is found via the Schrödinger equation [15]. Let us denote the pure quantum state that our parameterized quantum circuit makes at some initial time t is $|\psi(\boldsymbol{\theta}(t))\rangle = \boldsymbol{\psi}(\boldsymbol{\theta}(t))$, then the (time-independent) Schrödinger equation is

$$\frac{d}{dt}|\psi(\boldsymbol{\theta}(t))\rangle = -iH|\psi(\boldsymbol{\theta}(t))\rangle, \tag{3.18a}$$

$$\frac{d}{dt}\boldsymbol{\psi}(\boldsymbol{\theta}(t)) = -i\mathbf{H}\boldsymbol{\psi}(\boldsymbol{\theta}(t)). \tag{3.18b}$$

The equation describes the infinitesimal change of a quantum state when the spins are acted upon by some energy operator H which generally describes the potential U and kinetic V energies of the system, that is, $H = U + V$. We shall throughout this thesis only investigate time-independent Hamiltonians where the kinetic energy V is zero (such models for spins on a static lattice) and the potential energy U (such as is the case for all the Hamiltonians we considered in Section 3.2). We emphasize, that what is evolving over time in Eq. (3.18) is the parameters $\boldsymbol{\theta}(t)$ that produces our quantum state $|\psi(\boldsymbol{\theta}(t))\rangle = \boldsymbol{\psi}(\boldsymbol{\theta}(t))$. The variational principle [217] lets us to map the time evolution of the quantum state to time evolution of the parameters [225]. From this principle, one can derived how the parameters should evolve in order to simulate the Schrödinger evolution (Eq. (3.18)).

Despite simulating time evolution is an interesting and important problem in itself, a very peculiarly thing happens if we substitute t to be $\tau = it$ in Eq. (3.29), that is, creating *imaginary time evolution* (ITE) [226]

$$|\psi(\boldsymbol{\theta}(\tau + \delta\tau))\rangle = e^{-\tau H}|\psi(\boldsymbol{\theta}(t))\rangle, \tag{3.19a}$$

$$\boldsymbol{\psi}(\boldsymbol{\theta}(\tau + \delta\tau)) = e^{-\tau\mathbf{H}}\boldsymbol{\psi}(\boldsymbol{\theta}(t)). \tag{3.19b}$$

Although, it is an unphysical process, it is often used as a mathematical tool to solve problems more generally in physics. ITE yields the following update equation to the parameters $\boldsymbol{\theta}(t)$ over a small time step $\delta\tau$

$$\boldsymbol{\theta}(\tau + \delta\tau) \simeq \boldsymbol{\theta}(\tau) + \delta\tau\mathbf{A}^{-1}(\tau)\mathbf{c}(\tau) \tag{3.20}$$

which turns out to be exactly the same update rule as we saw with natural gradient descent, since $\mathbf{c}(\tau)$

is the gradient and $\mathbf{F} = \mathbf{A}$ [227, 228]. However, it is only equivalent for pure states evolving in closed systems. As Koczor et al. shows in [229], density matrices evolving in non-unitary circuits, one cannot use the *classical* Fisher information; but should instead use the quantum Fisher information. Whereas the classical Fisher information describes how the probability distribution $p(i|\boldsymbol{\theta})$ changes when the parameters $\boldsymbol{\theta}$ are changed, the quantum Fisher information describes how the quantum state $\rho(\boldsymbol{\theta})$ changes. The quantum Fisher information matrix is, however, more difficult to estimate and there exist multiple approaches, including using VQAs to do so [230]. One estimate from ref. [229] is given by

$$[\mathbf{F}_Q(\boldsymbol{\theta})]_{kl} \simeq \mathrm{Re}\left\{\mathrm{Tr}\left[\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial\rho(\boldsymbol{\theta})}{\partial\theta_j}\right]\right\} \tag{3.21}$$

which reduces to $\mathbf{F}$ when $\rho(\boldsymbol{\theta})$ is pure.

When our estimate $\boldsymbol{\theta}$ is in the vicinity of a local minimum, Koczor et al. showed that there was an analytical solution to the optimization problem due to the periodicity in the parameters and the nature of the cost function [231]. Other encouraging ideas include measuring covariances that yields update rules to our circuit parameters which has shown great promise for VQE type problems, especially when we are in the neighborhood of a minimum [232]. On the classical optimizer side of VQAs, there has been extensive investigation of what works best. It is often argued that gradient-free methods as well as plain gradient descent is not sufficient as they are easily stuck in local minima and even popular methods in classical machine learning such as the Adam optimizer [233] does not scale well with qubit count [234]. It is generally argued that these quantum natural gradient descent methods outperform standard gradient descent [61]. Another big question was also whether VQAs could be used in aiding QEC, and indeed they could [235], which creates hope in the near future for a smoother transition to fault-tolerance.

3.3.1 Ansatz

The *ansatz* (plural: ansätze) is another word for the circuit unitary $U(\boldsymbol{\theta})$. Since $U(\boldsymbol{\theta})$ constraints which quantum states our circuit can produce, or the *expressiveness* of the circuit, there has been put extensive research into the design of the circuits in order to be expressive and still limit circuit depth. Ref. [236] defines expressivity as the quantum circuit's ability to generate states uniformly over all possible quantum states. They also investigate the *entangling capability* which they measure through the Meyer-Wallach (MW) entanglement metric due to its scalability and ease of computation. With these two metrics they simulate a collection of circuit templates to see which template(s) are superior relative

to their size and depth.

A particular type of ansätze known as hardware-efficient ansätze [237] is a family of unitary gates easily implementable on quantum hardware aiming at reducing circuit depth and thus minimizing the potential amount noise that can be accumulated. However, it turns out that these can be harder to train since a good solution to the problem might not lie in the hypothesis class $U(\boldsymbol{\theta})$. Another line of research have investigated Hamiltonian Variational Ansätze (HVA) which creates the ansatz according to the Hamiltonian of interest [238]; very similar to QAOA which alternates between unitaries of initial and problem Hamiltonian [107]. We saw with the adiabatic theorem (see Section 2.3.1) that it was possible to Trotterize the analog adiabatic pathway, and using this idea, the layers in HVA corresponds to one such discrete step. There also exists more exotic ansätze such as adaptively add/remove gates and thus optimizing circuit structure [239]. It has also been shown that parameter initialization $\boldsymbol{\theta}^{(0)}$ is crucial in order to converge to a good solution [240]. Meta-learning protocols such as using a classical neural network to control the circuit has here been proposed [162].

In 2018, McClean et al. [241] discovered that in the training landscape of QNNs there was a "vanishing gradient" like problem coined *barren plateaus* (BP). As the number of qubits increase, the size of the wave-vector space increases exponentially and so too does the numerical size of the gradients decrease exponentially [202]. This key result shifted the field of QNN quite a bit, as research was now trying to come up with tactics to avoid or minimize the effect of BPs. In the light of BPs, subsequent papers showed that there is an inherent balance between expressability and trainability in QNNs [242]. Here, trainability refers to expected gradients vanishing at most polynomially in system size. Moreover, it is crucial when designing PQCs that the ansatz is *complete*, that is, it contains a good solution to the problem Hamiltonian. Not only can the ansatz and bad initialization lead to BPs, but there also exists noise-induced BPs which makes QNNs extra hard to train on actual hardware [124]. The convergence results when noise is present is related to the quantum Fisher information [243]. Some results indicate that BPs can be avoided for specific architectures such as quantum convolutional neural networks [244] and exploiting parameter correlation can also lead to larger gradients [245]. However just as for classical neural networks, training QNNs is NP-hard [246] given that there is exponentially many local minima [247]. Recently, Anscheutz argued that there are for many QNNs trainabilty problems beyond BPs [169]. Cerezo et al. [248] showed that the cost function—i.e. the Hamiltonian—structure is also crucial for the existence of BPs. They show that circuits finding performing VQE for local Hamiltonians (the one we consider in this thesis) are trainable as long as the circuit-depth scales $\mathcal{O}(\log N)$.

Lately, a lot of research has been inspired by geometric deep learning by exploiting symmetries in the data [249]. When comparing to classical neural networks, QNNs have in some instances been shown to outperform in generative tasks [250]. However, a crucial element in this expressive advantage is how the data is encoding in variational quantum-machine-learning models [251].

3.4 Estimating Expectation Values

One very critical aspect of VQAs is the number of measurements (also called shots) required to obtain accurate and sufficient measurement statistics and subsequently update the cost function. This is often argued to one of the mayor bottlenecks of practical VQA applicability [252, 253]. In Section 3.3.1, we saw the phenomenon of Barren Plateaus (Section 3.3.1) when computing gradients. Here we emphasize that if the gradients are smaller than the error in the expectation due to insufficient amount of shots, we risk getting the wrong sign of the gradient which leads to a random walk [254]. Thus we would like to know how many shots are required to get reliable gradients, and this turns out to be a non-trivial task.

Recall that when we optimize QNNs w.r.t. parameters $\boldsymbol{\theta}$, we need an estimate of the expectation value of some Hermitian operator given a quantum state,

$$\langle \mathcal{H}(\boldsymbol{C}) \rangle = \mathrm{Tr}[\rho(\boldsymbol{\theta})\mathcal{H}(\boldsymbol{C})], \tag{3.22}$$

where $\mathcal{H}(\boldsymbol{C}) = \sum_{m=1}^{M} c_m P_m$ is a general k-local Hamiltonian with coefficients $\boldsymbol{C} = \{c_1, c_2, ..., c_M\}$ and $\rho(\boldsymbol{\theta})$ is the density matrix our parameterized quantum circuit produces. We do this experimentally by computing the expectation value of each $\langle P_m \rangle = \mathrm{Tr}[\rho(\boldsymbol{\theta})P_m]$ and then subsequently plugging this into $\langle \mathcal{H}(\boldsymbol{C}) \rangle$.

We can now derive the variance of each estimator $\langle P_m \rangle$ in $\mathcal{H} = \sum_{m=1}^{M} c_m P_m$ given N_{shots} samples. Measuring the i'th spin at the end of our circuit, results in a binary outcome $z_i \in \{-1, 1\}$. The expectation $\langle z_i \rangle := \mathbb{E}[z_i] \in [-1; 1]$ is itself a random variable. At large sample sizes ($N_{shots} > 1000$) this expectation is well approximated by a normal distribution $\mathbb{E}[z_i] \sim \mathcal{N}(\mu, \sigma^2)$ due to the central limit theorem [145]. The standard error of the mean [255], denoted ϵ, scales with $\epsilon = \mathcal{O}\left(\frac{1}{\sqrt{N_{shots}}}\right)$, and thus the number of samples required to obtain precision ϵ scales inversely $N_{shots} = \mathcal{O}\left(\frac{1}{\epsilon^2}\right)$. Having M terms $\langle P_m \rangle$ in our Hamiltonian, each needing $\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$ samples, the total number of shots to estimate the expectation in Eq. (3.22) scales as $\mathcal{O}(\frac{M}{\epsilon^2})$. We note that this is irrespective of if the state $\rho(\boldsymbol{\theta})$ is pure or

mixed, albeit as we shall see momentarily, noise (as might be expected) increases N_{shots} even further.

Generally, some of the terms in P_m are likely to *commute* with other terms. Hence measuring P_u does not affect the expectation value of another term P_v as the wavefunction only collapses the parts we measured. Thus there is possibility for clever grouping the commuting terms and *not* having to repeat the entire circuit [256] for each term. In fact, Huggins et al. provided a measurement strategy such that the total number of measurements for a Hamiltonian expectation scales $\mathcal{O}(\frac{\sqrt{M}}{\epsilon})$ even for non-commuting terms P_m, however, with the price of increased circuit depth linear in the number of qubits and using an additional $\mathcal{O}(M \log \frac{1}{\epsilon})$ qubits [257]. Another highly cited approach is so-called *classical shadows* which introduces random projections just before measurement to measure in the X, Y or regular Z basis [258]. This randomness allows one to compute a classical shadow—a classical representation of the quantum state from its tomography—that can be used to estimate expectations in $\mathcal{O}\left(\frac{\log M}{\epsilon^2}\right)$ runs and it has shown to scale better asymptotically with the number of spins [259]. One can also formulate the grouping of observable as a minimum-clique graph problem to obtain more efficient measurement protocols [260]. Another interesting proposal is to linearly combine unitary (LCU) operators in order to measure a group of fully anticommuting terms of the Hamiltonian in a single series of single qubit measurements [261].

Although it has been argued that VQAs have a scalability problem due to the number of samples required to estimate expectations [253], other ideas such as multi-core quantum computing [262] proposes the solution of distributing what can be parallelized, namely the repetitions/samples of the forward-passes of quantum circuits. As there has been provable advantage for large scale data [263] and we know theoretically that there is quantum advantage in learning from experiments [264], it seems logical for the research field keep searching for solutions to these near-term problems, the biggest of which arguably is the noise in NISQ devices. A big issue originating from the nature of the noise is that we are not necessarily guaranteed we can just repeat the experiment enough times to get an accurate estimate of $\langle \mathcal{H}(\boldsymbol{C}) \rangle$. In general, noise might introduce a *bias* in the estimate. In order to limit this bias and how many extra samples are required to estimate the expectation value, the field of *quantum error mitigation* is introduced in the next section.

3.5 Error Mitigation

Quantum error mitigation (QEM) is a relatively new sub-field in quantum computing aiming at understanding how noise affects the expected value of measurement operators and how one can mitigate

these effects. QEM is the "cheap" version of QEC in that it is a collection of techniques aiding in computing more accurate expectation values at some cost such as requiring extra number of qubits, increased circuit depth or extra measurements. Compared to QEC needing poly-logarithmic more qubits and circuit depth [42], QEM does not correct errors in the quantum state, that is, it lets the quantum state decohere and only focuses on estimating expectation values of operators. QEM often either assumes a specific noise model or approximates the noise in the quantum computer. Thus QEM does not guarantee errors in expectation to be suppressed unless the noise model or approximations are accurate. It is hard to verify if the noise model is accurate (some QEM techniques try this) and the noise might change over the course of multiple experiments as the hardware heats up.

In general, our goal is to accurately estimate $\langle \mathcal{H} \rangle = \mathrm{Tr}[\rho(\boldsymbol{\theta})_{id}\mathcal{H}]$ (omitting the Hamiltonian parameters $\boldsymbol{C}$ for brevity) for some ideal, noise-free, state $\rho(\boldsymbol{\theta})_{id}$. However, our circuit produces a noisy version $\rho(\boldsymbol{\theta})$ and we can only estimate $\langle \mathcal{H} \rangle$ with finite measurements and so in practice we only obtain an estimate $\langle \hat{\mathcal{H}} \rangle$ by running the circuit N_{shots} times. To emphasize, there are two sources of error: from the fact that $\rho(\boldsymbol{\theta})_{id} \approx \rho(\boldsymbol{\theta})$ and that $N_{shots} < \infty$. Moreover, in general, the effect of the noise is that we obtain a **biased** estimate $\langle \hat{\mathcal{H}} \rangle$; to see why, we can ask: what is the average (squared) difference between $\langle \hat{\mathcal{H}} \rangle$ and $\langle \mathcal{H} \rangle$? QEM techniques all aim at minimizing the mean square error between them, that is, minimizing

$$\mathrm{MSE}(\langle \hat{\mathcal{H}} \rangle) = \mathbb{E}\left[(\langle \hat{\mathcal{H}} \rangle - \langle \mathcal{H} \rangle)^2\right]. \tag{3.23}$$

This is a very general problem and the above loss function is one of the most used regression losses in machine learning [52]. Machine learners thus know that the MSE can be decomposed into the *bias* and the *variance* (expanding the product) yielding

$$\mathrm{MSE}(\langle \hat{\mathcal{H}} \rangle) = \underbrace{\left(\mathbb{E}\left[\langle \hat{\mathcal{H}} \rangle\right] - \langle \mathcal{H} \rangle\right)^2}_{bias} + \underbrace{\mathbb{E}\left[\langle \hat{\mathcal{H}} \rangle^2\right] - \mathbb{E}\left[\langle \hat{\mathcal{H}} \rangle\right]^2}_{variance} \tag{3.24}$$

If we denote $\langle \hat{\mathcal{H}} \rangle_{N_{shots}}$ to be the estimator after running the circuits N_{shots} times and thus averaging N_{shots} expectation values, its mean square error is

$$\begin{aligned}\text{MSE}\left(\langle\hat{\mathcal{H}}\rangle_{N_{shots}}\right) &= \left(\mathbb{E}[\langle\hat{\mathcal{H}}\rangle_{N_{shots}}] - \langle\mathcal{H}\rangle\right)^2 + \frac{1}{N_{shots}}\left(\mathbb{E}\left[\langle\hat{\mathcal{H}}\rangle^2_{N_{shots}}\right] - \mathbb{E}\left[\langle\hat{\mathcal{H}}\rangle_{N_{shots}}\right]^2\right) \\ &= \left(\text{Tr}[\rho(\boldsymbol{\theta})\mathcal{H}] - \text{Tr}[\rho(\boldsymbol{\theta})_{id}\mathcal{H}]\right)^2 + \frac{1}{N_{shots}}\left(\text{Tr}\left[\rho(\boldsymbol{\theta})\mathcal{H}^2\right] - \text{Tr}[\rho(\boldsymbol{\theta})\mathcal{H}]^2\right).\end{aligned} \tag{3.25}$$

Increasing the number of shots does nothing to the bias; it only decreases the variance, which we refer to as *shot noise*. The goal of QEM is to limit the bias term, typically at the cost of increasing the number of shots, qubits and/or circuit depth.

When the circuit state is different from the ideal state—such as when affected by noisy— that is, $\rho(\boldsymbol{\theta}) \neq \rho(\boldsymbol{\theta})_{id}$ getting expectation values to precision ϵ generally requires even more samples than in the pure case; how much depends on the circuit noise level as well as to which degree the noise affects the density matrix uniformly (white noise) or there is structure in the noise. We shall use the model of discrete single-qubit fault probability, that is, after every moment in our circuit (see example in Fig. 2.3), some error happens with probability ϵ to each qubit. Thus having a total of N qubits and M moments, there are $\nu = NM$ trials for an error to happen, and we can thus define the *circuit error rate* ξ as

$$\xi := \epsilon\nu. \tag{3.26}$$

The error rate ξ can also be seen as the expected number of total errors happening in one run. If errors are independent, it can be shown that the probability of no error occurring decays exponentially with ξ and thus the number of extra shots needed—assuming the pauli noise model (Section 2.2.1)—increases exponentially [265] and thus any potential speedup over classical attempts vanishes [61]. This exponential decay is easily shown for many noise models (such as the ones introduced in Section 2.2) that allow the decomposition

$$\rho = (1-\epsilon)\rho_{id} + \epsilon\rho_{err} \tag{3.27}$$

Having ν of such channels, yields

$$\rho = (1-\epsilon)^\nu\rho_{id} + (1-(1-\epsilon)^\nu)\rho_{err} \tag{3.28}$$

and thus an exponential decay $(1-\epsilon)^\nu$ with system size/depth. In fact, for many QEM techniques,

an exponential cost is unavoidable even if one increases the number of qubits or the circuit depth, and moreover, training of VQAs only in certain instances benefits from QEM [266].

For a noisy circuit, we can denote the initial quantum state $\rho_0 = \boldsymbol{\rho}_0$ and the corresponding output $\rho(\boldsymbol{\theta}, \epsilon) = \boldsymbol{\rho}(\boldsymbol{\theta}, \epsilon)$ is made by the overall CPTP map of the entire circuit

$$\rho(\boldsymbol{\theta}, \epsilon) = \Phi_c(\boldsymbol{\theta}, \epsilon)\rho_0, \tag{3.29a}$$

$$\boldsymbol{\rho}(\boldsymbol{\theta}, \epsilon) = \Phi_c(\boldsymbol{\theta}, \epsilon)\boldsymbol{\rho}_0. \tag{3.29b}$$

Here Φ_c is the circuit CPTP map that only approximately (due to possibly correlated noise) decomposes as

$$\Phi_c(\boldsymbol{\theta}, \epsilon) \approx \Phi_\nu(\theta_\nu, \epsilon)...\Phi_2(\theta_2, \epsilon) \cdot \Phi_1(\theta_1, \epsilon). \tag{3.30}$$

For many types of errors, we can write the combined CPTP map of a unitary gate (which could be the identity) followed by an error happening with probability ϵ, that is, for the k'th noisy quantum channel we have

$$\Phi_k(\theta_k, \epsilon)\rho = (1-\epsilon)U_k(\theta_k)\rho U_k(\theta_k)^\dagger + \epsilon \sum_j^J K_{jk}\rho K_{jk}^\dagger \tag{3.31}$$

where U_k is the unitary operation happening with probability $1-\epsilon$ and K_{jk} are the J Kraus operators modelling some noise process.

Koczor investigates how the mismatch between the *dominant eigenvector* $|\lambda_m\rangle = \boldsymbol{\lambda}_m$ of the density matrix $\rho(\boldsymbol{\theta})$ and the corresponding ideal pure state $\rho_{id}(\boldsymbol{\theta}) = |\psi_{ideal}(\boldsymbol{\theta})\rangle\langle\psi_{ideal}(\boldsymbol{\theta})|$ the circuit would produce if it was noise-free [267]. This is an interesting problem since what we are interested in is exactly the expected energy of the state for zero noise, and thus quantifying the mismatch is central to understand how noise affects our expectations. Writing the density matrix in its spectral decomposition (see Eq. (2.11)),

$$\rho(\boldsymbol{\theta}) = \lambda_m \cdot |\lambda_m\rangle\langle\lambda_m| + \sum_{i=2}^{2^N} \lambda_i |\lambda_i\rangle\langle\lambda_i|, \tag{3.32a}$$

$$\boldsymbol{\rho}(\boldsymbol{\theta}) = \lambda_m \cdot \boldsymbol{\lambda}_m\boldsymbol{\lambda}_m^H + \sum_{i=2}^{2^N} \boldsymbol{\lambda}_i\boldsymbol{\lambda}_i^H. \tag{3.32b}$$

we can extract the eigenvector with the highest eigenvalue from the sum (note the sum starts at $i = 2$), and regard the remaining orthogonal states as noisy states arising from some non-trivial noise process. In the special case, when the noise ρ_{err} commutes with ρ_{ideal}, we know $|\lambda_m\rangle\langle\lambda_m| = \rho_{id}(\boldsymbol{\theta})$, however, incoherent noise is not likely—an in fact very unlikely—to commute with the ideal state and thus introduce mismatch between these two vectors. Koczor argues, however, that for sufficiently complex circuits, such as the QNNs often used in quantum machine learning, the dominant eigenvector is a very good approximation to the ideal state. In fact, it is possible to upper bounds the mismatch as defined by the infidelity (one minus the fidelity in Eq. (2.42)) between the dominant eigenvector and the ideal pure state $c := 1 - |\langle\psi_{ideal}(\boldsymbol{\theta})|\lambda_m\rangle|^2$, to be

$$c \leq (1 - \sqrt{1 - \delta^2})/2 \tag{3.33}$$

where $\delta := \lambda_m(\epsilon^{-1} - 1)$ i.e. the largest eigenvalue of the error terms.

One of the first QEM proposals was the zero-noise extrapolation [268] which considers the parameterized quantum circuit produced state $\rho_\xi(\boldsymbol{\theta})$ at circuit error rate ξ. The corresponding loss function $\mathrm{Tr}[\rho_\xi(\boldsymbol{\theta})\mathcal{H}]$ is then considered a function of ξ, where it is assumed that the hardware can each some minimum circuit rate but can always increase $\xi \geq \xi_{min}$. In fact it is fairly straightforward to increase the circuit error rate, since the experimentalist would just let more time pass between operations, or to spend more time applying gates, and thus letting more errors occur. The zero-noise extrapolation is thus a method where a function $f(\xi) = \mathrm{Tr}[\rho_\xi(\boldsymbol{\theta})\mathcal{H}]$ is fitted to various circuit error rates $[\xi_{min}, \xi_1, \xi_2, ..., \xi_k]$ and then one can extrapolate back to the expectation at zero circuit noise noise $f(0) = \mathrm{Tr}[\rho_0(\boldsymbol{\theta})\mathcal{H}]$.

As the Google team showed experimentally [166] and Dalzell et al. showed analytically [269], some types of quantum circuits scrambles local one-qubit noise into global white noise on the overall quantum state $\rho(\boldsymbol{\theta})$. For these white-noise generating circuits, error mitigation can be done but comes at exponential cost in system size [270]. As there is not a contributing bias term in the MSE loss (see Eq. (3.24)), i.e. only a rescaling factor, we get

$$\mathrm{Tr}[\rho(\boldsymbol{\theta})_{id}\mathcal{H}] = \frac{\mathrm{Tr}[\rho(\boldsymbol{\theta})\mathcal{H}]}{\eta} \tag{3.34}$$

for $\eta := (1 - \epsilon)^\nu$ where η can be estimated experimentally [271]. However, as we address in Paper B (Chapter 5) this global depolarization model is not necessarily a good approximation for all types of NISQ architectures to how local noise accumulates into the quantum state yielding the need for more

advanced QEM techniques, one of which is Virtual Distillation (VD) [272] / Error supression by derangements (ESD) [273]. The idea of VD/ESD is to prepare n copies of the state $\rho(\boldsymbol{\theta})$, entangle the copies and then compute the expectation on the overall state $\mathrm{Tr}[\rho(\boldsymbol{\theta})^n \mathcal{H}]$. This guarantees exponential decay of bias with n while still being NISQ friendly [274] by not needing that many copies ($n = 4$ show promising performance in the original paper).

Chapter 4

Paper A: Noise-Assisted Variational Quantum Thermalization

4.1 Foreword

The first paper presented is

- **J. Foldager**, A. Pesah, and L.K. Hansen. Noise-assisted variational quantum thermalization. *Scientific reports*, 12(1):1–11, 2022 [1]

which can be seen in its full extend in Appendix A [1]. This is joint work with Arthur Pesah from University College London in 2020-2021. The project started out with my first independent research idea which was using imaginary time evolution to approximate a thermal state and use this to train a restricted Boltzmann machine (RBM, see Section 2.9). While working a few months on this idea, a paper came out [275] doing *exactly* the same and it even had slightly better performance than my simulations. Despite being bummed out several weeks of the summer of 2020, we turned the idea into manipulating noise channels and learning these noise parameters together with unitary parameters in order to prepare a thermal state of a Hamiltonian. The following section includes a short summary of main findings in the paper and subsequently a section on the applications of the algorithm for restricted Boltzmann machines is provided in Section 4.3.

[1] This article is licensed under a Creative Commons Attribution 4.0 International License.

4.2 Summary

Variational Quantum Thermalization (VQT) is a subclass of Variational Quantum Algorithms (VQAs, see Chapter 3) which aims at putting qubits in a thermal state (Eq. (2.62)) that subsequently can be used for sampling tasks. This is often by either using twice as many qubits $2N$, entangle them pairwise and only caring about the reduced state on N qubits, or choosing a mixed input state ρ_{init}.

In ref. [1], we go beyond parameterizing the unitary components of the circuit and include parameterization the noise channels as well. Specifically, we use the fact that the depolarization channel has the decomposition

$$D_\lambda(\rho) = (1-\lambda)\rho + \lambda d^{-1}\mathbb{1}. \tag{4.1}$$

This single-qubit depolarization channel is used after each qubit, and this is what essentially pumps in entropy to our system by a convex combination of the input state and the maximally mixed state. Moreover, we assume the probability λ to be constant across all qubits and layers, and include this noise parameter as one of the learnable parameteres when minimizing the free energy (see Section 2.5 approximates the thermal state). This protocol we coin *Noise-assisted variational quantum thermalization* (NAVQT).

In order to use the free energy (Eq. (2.59)) as a loss function, we derive an approximation to the entropy of the quantum state by shifting all depolarization channels to the beginning of the circuit (see Fig. 1 in the paper). Given that the entropy does not change with unitary operations, all entropy in the system is introduced in the beginning and thus leaving our entropy approximation independent of the unitary circuit parameters $\boldsymbol{\theta}$. The free energy approximation thus becomes,

$$F(\boldsymbol{\theta},\lambda) \approx \mathcal{L}(\boldsymbol{\theta},\lambda) := \mathrm{Tr}[\rho(\boldsymbol{\theta},\lambda)\mathcal{H}] - \beta^{-1}\tilde{S}(\lambda) \tag{4.2}$$

where the approximate entropy $\tilde{S}(\lambda)$ is derived in the paper and given by

$$\begin{aligned}\tilde{S}(\lambda) = -N&\left((1-\lambda)^m + \frac{(1-(1-\lambda)^m)}{d}\right)\cdot \\ &\ln\left((1-\lambda)^m + \frac{(1-(1-\lambda)^m)}{d}\right) + \frac{(d-1)(1-(1-\lambda)^m)}{d}\ln\left(\frac{(1-(1-\lambda)^m)}{d}\right)\end{aligned} \tag{4.3}$$

for a circuit with m unitary layers (or it could be moments) and N qubits. The loss function $\mathcal{L}(\boldsymbol{\theta},\lambda)$ has gradients for all parameters and thus we can use gradient based method to optimize our quantum

computer. We numerically validate in the supplementary material that the true free energy $F(\boldsymbol{\theta}, \lambda)$ follows the approximation very closely.

We test the algorithm for a range of different 1D spin chain models including the Ising chain, transverse-field Ising chain and Heisenberg model as Hamiltonians $\mathcal{H}$ in Eq. (4.2). We find that for a range of temperatures, the algorithm nicely converges for all three Hamiltonians but the fidelity with the true thermal state depends on the temperature. In the next section, we shall see how one can use NAVQT to train a restricted Boltzmann machine (see Section 2.9).

Correction to the original paper In the original paper, it says that the dimensionality in our entropy approximation $\tilde{S}(\lambda)$ is $d = 2^N$. However, this should be $d = 2$, since we derive the the entropy approximation as being the sum of N (independent) entropy terms, where each term is the accumulated entropy happening due to single-qubit depolarization noise living in a two dimensional Hilbert space and not in an 2^N dimensional space. See the supplementary material. See Appendix A for detailed derivation but note that d should be 2.

4.3 Detecting Speech Patterns

As a proof of concept, an experiment detecting patterns in speech signals is provided to illustrate the applicability of being able to sample from a thermal state.

Recall that the Hamiltonian (energy operator) of the restricted Boltzmann machine and corresponding probability distribution is the thermal state given by

$$H := H(\mathbf{x}_v, \boldsymbol{z}_h) = -\sum_{h=1}^{N_h} b_h z_h - \sum_{v=1}^{N_v} b_v x_v - \sum_{h=1}^{N_h}\sum_{v=1}^{N_v} w_{vh} z_h x_v \tag{4.4}$$

$$p_\beta(\mathbf{x}_v, \boldsymbol{z}_h) = \frac{1}{Z} e^{-\beta H} \tag{4.5}$$

that is, our Hamiltonian is the Ising model. Research has also tried out Hamiltonians with non-diagonal elements such as the transverse-field Ising models performing better than comparable classical models [276], but for illustration purposes we stick with the classical Ising model. Training restricted Boltzmann machines (RBMs) can be done by minimizing the average negative log-likelihood. Thus requires one to compute the gradient δ of each parameter in the RBM, that is,

$$\delta\theta_j = \sum_{n=1}^{N_{train}} (\mathrm{Tr}[\partial_j H_c \rho_{c_n}]) - \mathrm{Tr}[\partial_j H_u \rho_u], \tag{4.6}$$

for N_{train} datapoints. Here ρ_{c_n} is the thermal state of the Hamiltonian H_{c_n} when the visible neurons are *clamped* to the n'th datapoint and ρ_u is the thermal state of the *unclamped* Hamiltonian H_u, given by

$$H_c := -\sum_{v=1}^{N_v} b_h Z_h - \sum_{h=1}^{N_h}\sum_{v=1}^{N_v} w_{vh} Z_h Z_v \tag{4.7}$$

$$H_u := -\sum_{v=1}^{N_v} b_h Z_h - \sum_{v=1}^{N_v} b_v Z_v - \sum_{h=1}^{N_h}\sum_{v=1}^{N_v} w_{vh} Z_h Z_v \tag{4.8}$$

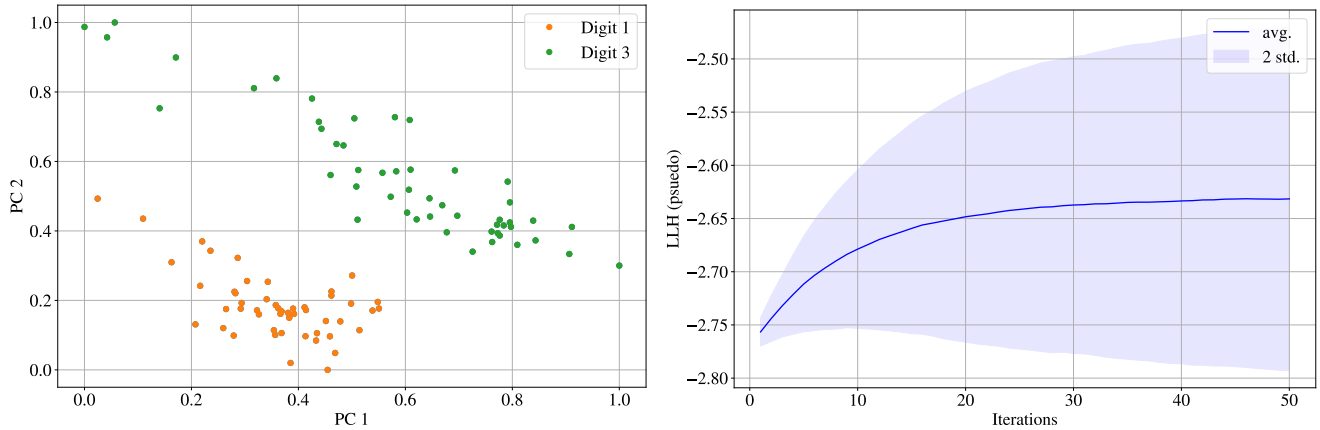
where Z is the Pauli Z operator acting on a single qubit. Given the simulatable restrictions of NISQ devices, the RBM in these experiments will contain $N_v = 4$ visible and two hidden $N_h = 2$ neurons, that is, a total of $N = 6$ qubits. Since $\partial_j H$ only keeps the term where the j'th parameter is in the Hamiltonian, the gradients for the biases and weights become, respectively

$$\delta b_k := \sum_{n=1}^{N_{train}} \mathrm{Tr}[Z_k \rho_{c_n}] - \mathrm{Tr}[Z_k \rho_u], \tag{4.9}$$

$$\delta w_{ij} := \sum_{n=1}^{N_{train}} \mathrm{Tr}[Z_i Z_j \rho_{c_n}] - \mathrm{Tr}[Z_i Z_j \rho_u]. \tag{4.10}$$

We see that in order to do gradient descent for weights and biases in a RBM, we need to compute expectation values of observables under some thermal state. Preparing the thermal state is exactly what NAVQT does and a job which quantum computers is thought to beat classical computers in [277, 278].

The Audio MNIST [279] is a dataset of spoken digits from 0-9 by 60 different speakers each pronouncing a digit 50 times. Using the last transformer layer from Wave2Vec2 [280] and Principal Component Analysis (PCA) to get four dimensional embeddings of the recorded speech signal, we now make a small proof-of-concept experiment where the job of the RBM is learn to classify between two digits (1 and 3). The result is displayed in Fig. 4.1 and we see that it is indeed possible to train a RBM to classify the samples, albeit it should be straightforward for many models.



(a) **(b)**

Figure 4.1: Training a Restricted Boltzmann Machine for Speech Classification. We use two hidden neurons and four visible units of the first four principal components. **(a)** Data projected onto the two first principal components of the last layer Wave2Vec2 embeddings. **(b)** The pseudo log likelihood (LLH) as a function of optimization iterations averaged over one hundred seeds. A mean test accuracy of 0.937 is obtained with 95% confidence interval $[0.931, 0.943]$.

Chapter 5

Paper B: Can shallow quantum circuits scramble local noise into global white noise?

5.1 Foreword

The next paper presented is

- **J. Foldager** and B. Koczor. Can shallow quantum circuits scramble local noise into global white noise? *arXiv preprint arXiv:2302.00881*, 2023 [2]

which can be seen in its full extend in Appendix B. This is joint work with Balint Koczor from University of Oxford carried out in the spring of 2022. The paper studies how noise accumulates in specific types of NISQ quantum circuits relevant for quantum machine learning (QML).

5.2 Summary

The local Pauli Error noise Model (PEM) assumes that after each moment (or unitary operation) in the circuit, every qubit is hit by a noise channel which can be described as a sum of Pauli gates. The resulting noisy state ρ produced by the circuit depends on both which unitary gates are in the circuit as well as which errors affected it. A previous result by Dalzell et al. [269] showed that random quantum circuits transform local noise into global white noise, but before our contribution it was not clear if

popular NISQ architectures often used in QML also does this. Since several QEM proposals works well with white noise affected quantum states [281], it is not clear for which types of architectures this assumption does not hold and how to get unified metrics that investigates this. This is what the paper is about, as we propose two metrics given by

$$W := \frac{1}{2}\|p_{err} - p_{unif}\|_1 = \frac{1}{2}\sum_{k=2}^{d}\left|\frac{\lambda_k}{1-\lambda_m} - \frac{1}{d-1}\right|, \tag{5.1}$$

$$C := \frac{||\,[\rho_{ideal}, \rho]\,||_1}{1-\lambda_m} = \|[\rho_{ideal}, \rho_{err}]\|_1 + \mathcal{O}(\mathcal{E}_a). \tag{5.2}$$

where $\mathcal{E}_a$ is some approximation error. Here $d = 2^N$, p_{err} is the spectrum of the density matrix ρ without the dominant eigenvalue λ_m (see Section 3.5) . Whereas W (being the l_1 distance to white noise) describes how well the collection of non-dominating eigenvalues approximates white noise, C describes the commutator norm between the ideal state ρ_{id} and the circuit produced state ρ such that the dominating eigenvalue λ_m of ρ approximates the fidelity $F(\rho_{id}, \rho)$

$$\lambda_m = F(\rho_{id}, \rho) + \mathcal{O}(||\,[\rho_{ideal}, \rho]\,||_1). \tag{5.3}$$

We find that in most cases white noise is not a good approximation to the eigenspectra of ρ. Instead, we find that both W and C as a function of the number of gates in the circuit is approximated well by

$$f(\nu) = \alpha\frac{e^{-\xi}\xi}{(1-e^{-\xi})\sqrt{\nu}} = \frac{\alpha}{\sqrt{\nu}} + O(\xi), \tag{5.4}$$

The factor α, we find, initially grows with N but then saturates and then not increasing N for sufficiently large systems. We simulate often used cost functions in QML including the Heisenberg XXX spin model, transverse-field Ising model and the Lithium Hydride (LiH) Hamiltonians. We investigate both the strong entangling layer and Hamiltonian variational ansatz (see Chapter 3) and find that when we initialize the circuit parameters randomly, we get a reasonably good fit as the number of gates are increased. However, when the parameters are not random such as when we approximate the adiabatic path, the HVA ansatz produces non-dominating eigenvalues in the density matrix not well approximated by white noise. However, it appears that the commutator norm C is sufficiently small and decreases as the number of gates becomes sufficielty large. As also explained in the paper, we argue that given the HVA depends

on the problem Hamiltonian, the circuit might not have sufficiently large Lie algebra, i.e., limited ability to scramble noise into white noise, which explains the low white noise similarity. By inserting additional gates into the circuit which increases the Lie algebra, we see significantly smaller C metric and a clear decrease with the number of gates.

Given we find that popular QML architectures does not in general produce a dominating eigenvector close to the ideal state plus white noise, this implies that using QEM techniques relying on this assumption might not be ideal. That is, if one applies QEM together with particular NISQ architetures (strong entangling layer and Hamiltonian variational ansatz) to mitigate noise effects, one should consider estimating metrics like C and W as these reveal efficacy about which QEM techniques will yield accurate and robust results.

Chapter 6

Paper C: Actively Learning Quantum Machine Learning Architectures from Related Problems

6.1 Foreword

The next paper presented is

- **J. Foldager**. Actively learning quantum machine learning architectures from related problems. 2023 [3]

which can be seen in its full extend in Appendix C. The paper develops a meta-learning algorithm that uses both Active learning (AL) and Bayesian optimization (BO) in order to learn quantum circuit architectures across various Hamiltonians.

6.2 Summary

The variational quantum eigensolver (VQE) aims at learning the parameters $\boldsymbol{\theta}^*$ in a quantum circuit $U(\boldsymbol{\theta})$ such that the output of the circuit is the groundstate of a Hamiltonian, that is,

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \langle \mathcal{H}(\boldsymbol{C}) \rangle = \langle \psi(\boldsymbol{\theta}) | \, \mathcal{H}(\boldsymbol{C}) \, | \psi(\boldsymbol{\theta}) \rangle \,, \tag{6.1}$$

where $\mathcal{H}(\boldsymbol{C})$ is a *parameterized* 2-local Hamiltonian,

$$\mathcal{H}(\boldsymbol{C}) := \sum_{(\alpha_i,\alpha_j)} \sum_i J_{ij}^{(\alpha_i,\alpha_j)} \sigma_i^{\alpha_i} \sigma_{i+1}^{\alpha_j} + \sum_{\alpha_i} \sum_i b_i^{\alpha_i} \sigma_i^{\alpha_i} \tag{6.2}$$

for parameters $\boldsymbol{C} = \{\{J_{ij}^{(\alpha_i,\alpha_j)}\}, \{b_i^{\alpha_i}\}\}$. Aside from the circuit parameters $\boldsymbol{\theta}$, we also have the VQA hyperparameters which is the overall experimental design, i.e., *how* we design the circuit including parameters such as the classical optimizer learning rate, number of ansatz layers, etc. We can therefore expand the notation such that the circuit unitary $U_{\boldsymbol{\Theta}}(\boldsymbol{\theta})$ is conditioned on the hyperparameters $\boldsymbol{\Theta}$. An open question in quantum neural network research is how to design the ansatz and classical optimization loop, that is, design the hyperparameters. The general hypothesis that sparked the idea of this paper is that given different hyperparameters $\boldsymbol{\Theta}_a \neq \boldsymbol{\Theta}_b$ lead to different coverage of reachable states in the Hilbert space that quantum states live in. But there is another dimension as well: how quick the VQA converges to a solution $|\boldsymbol{\theta}\rangle$ as well as the quality of that solution, which we can define as the infidelity between $|\psi(\boldsymbol{\theta}\rangle$ and the groundstate $|\psi_0\rangle$ given by

$$\mathcal{I} = 1 - |\, \langle \psi_0 | \psi(\boldsymbol{\theta}) \rangle \,|^2. \tag{6.3}$$

If the infidelity is low (close to zero), it means that our circuit has found a good solution to the problem, since $|\, \langle \psi_0 | \psi(\boldsymbol{\theta}) \rangle \,|^2 \approx 1$ i.e. their inner product is close to unity. However, this is not the loss that the classical agent uses to optimize as we would have to be able to calculate the inner product with the true groundstate. Instead, we—as done in general VQE—use the energy as loss function.

In the paper, we propose a way of letting a classical meta-learning algorithm query a certain number of Hamiltonians using AL, that is, a collection $\boldsymbol{C}_{collection} = \{\boldsymbol{C}_1, \boldsymbol{C}_2, ..., \boldsymbol{C}_K\}$ and for each of them run BO in order to optimize the hyperparameters $\boldsymbol{\Theta}$. The agent thus have an inner BO loop and an outer AL loop, the former conditioning on a specific Hamiltonian and optimizing for hyperparameters and the latter of which searches in the Hamiltonian parameter space to find areas where the uncertainty in the groundstate energy is high. The agent is thus named the *actively learned bayesian optimized* (ALBO) VQE. However, the approach is very general and could in principle be expanded to other types

of problems as well.

The goal of training such agent is to being able to handle it a set of Hamiltonian coefficients (that is, a new problem) and then it returns the optimal set of hyperparameters (VQA design) that solves the groundstate problem in as few iterations as possible. Specifically, we consider the number of ansatz layers as well as the learning rate as being the two hyperparameters we wish to optimize and learn, and the transverse-field ising (TFI, see Section 3.2) model and Heisenberg chain (XXZ, see Section 3.2). The numerical results show that ALBO outperforms both random search as well as standard choices of hyperparameters such as having half as many layers as qubits and a fairly large learning rate in the Adam optimizer. This suggest that it is possible learning across problems and that one benefits from actively choosing which Hamiltonians to find optimal hyperparameters for.

Chapter 7

Paper D: On the role of uncertainties in Bayesian Optimization

7.1 Foreword

Finally, we present the paper:

- **J. Foldager**, M. Jordahn, L.K. Hansen, and M.R. Andersen. On the role of model uncertainties in bayesian optimization. *arXiv preprint arXiv:2301.05983*, 2023 [4]

which can be seen in its full extend in Appendix D. The paper addresses the role of uncertainty estimates in Bayesian Optimization (BO) through an extensive study of the relationship between the BO performance (regret) and uncertainty calibration. We provide both numerical and theoretical evidence that for why uncertainty calibration might be difficult to combine with BO.

7.2 Summary

In the BO literature, there is a general consensus that the surrogate uncertainties (see Section 2.8) are crucial for a good BO performance as most acquisition functions depends heavily on the uncertainty. Moreover, it is often argued that one of the reasons why Gaussian Processes (GPs) often come out

superior to other surrogate models is due better / calibrated uncertainty estimates. In fact, a recent study [282] argues directly that *re-calibrating* these uncertainties enhances BO performance, i.e., the BO routine reaches a better estimate of the global minimum.

The intuition for calibrated uncertainties, as we also write in the paper, is straightforward in classification tasks. If a model provides p % chance that a datapoint belongs to a class, then on average we would expect p % of very similar datapoints to *actually belong* to that class. For example, if 10 datapoints gets 80% probability of belonging to class A, then if the model is calibrated, 8 of those 10 samples indeed belong to class A. For regression tasks it is slightly more involved, but given that a calibrated regression model generates a prediction μ and uncertainty estimate σ, we would see p percent of the data lying inside a p percentile confidence interval of μ. In practice, this is done by binning $p \in [0, 1]$ and create a calibration curve which is the Expected Confidence Level versus the Observed Confidence Level. Subsequently one can test how far a given regression model is to $y = x$ using the mean squared error, and we call this error the expected calibration error (ECE). Re-calibrating a regression model, as suggested by Kuleshov et al. [283], is the process of taking a predictive model and changing the predictive distribution such that the ECE is minimized. Recalibration is often done in an outer validation loop such as in ref. [282] which uses a subset of the available datapoints. Our contribution investigates if this is meaningful to do both through extensive numerical experiments as well as theoretically.

Our numerical experiments suggest that there are correlations between BO performance and calibration level *across* models. That is, if one is handed two models, one being well calibrated and one not being well calibrated, we would expect the calibrated one to arrive at the best solution on average. However, the effect is not significant when controlling for the type of surrogate model, that is *within* each model. In other words, our experiments show that as long as a practitioner chooses models that have a fair calibration ability (such as Gaussian Processes and Deep Ensembles), it does not help on average to be better calibrated when doing BO.

We also show that re-calibration does not (in general) improve BO performance, and importantly, we establish a mathematical proof that at small samples (where BO typically operates due to expensive queries of the objective function), a large variance in the calibration curve can be expected. Thus if one re-calibrates using a small validation set, one is likely to change the predictive distribution based on noise more than signal. We emphasize this in Proposition 1 in the paper, and this proposition is independent of the type of surrogate model. In summary, we argue that re-calibration during BO might be difficult.

Chapter 8

Conclusion

SOME day, maybe 10-100 years from now, a quantum computer migth be able to break current bank-level encryption [284], accelerate drug discovery [285], and empower artificial intelligence (AI) [97]. But today is not that day. Today, and the next several years, we only have access to noisy intermediate-scale quantum (NISQ) computers. In Chapter 1, we listed three key aspects concerning NISQ computers relevance for machine learning, it is hopefully now clear the author aspired to bringing us one step closer to approaching these. Chapter 2 spend quite some effort introducing quantum computing to computer scientists with little to no experience with quantum physics, taking all the way from simple probability distributions such as coin flipping to the postulates of quantum physics (Section 2.1), quantum computing noise models (Section 2.2), spin systems (Section 2.3) and thermal states (Section 2.5). Using this handpicked selection of the key aspects of quantum information theory relevant for this thesis, Chapter 3 introduced state of the art quantum neural networks as being hybrid quantum-classical algorithms with a lot of potential, unknowns and obstacles. Lastly, the scientific contributions in Chapters 4-7 were presented and summarized which leaves us with one last job of concluding on the combined findings, and we do so by directly repeating and answering the three scientific goals listen in Chapter 1.

- **Develop new NISQ algorithms which can be used to accelerate subroutines in ML.** There are key tasks in machine learning which are computationally hard and where approximate methods are the only way to go. Sampling high dimensional probability distributions is one of these challenges and thus quantum machine learning methods have been proposed to put qubits—which when measured follow a specific high dimensional distribution—in a *thermal state*. We call this variational

quantum thermalization (VQT), and the first paper, Paper A in Chapter 4, showed that by *learning* the noise in the quantum device, one can prepare several types of thermal states using only as many qubits as neurons in a restricted Boltzmann machine (RBM). The thermal state, we also showed, can be used to sample the expectation values used in the gradients of the RBM weights. Thus we have provided a novel way of training a RBM using a quantum computer

- **Gain a deeper understanding of how to characterize the unavoidable noise accumulation for NISQ algorithms.** We still do not know when NISQ computers will be practically applicable. We do know, however, that when we run quantum neural networks, there is a sampling overhead when one wishes to obtain accurate expectation values of observables. For this, it is crucial to use appropriate quantum error mitigation (QEM) techniques and Paper B in Chapter 5 proposes two metrics which can help to decide which QEM protocol to follow. We justify this both numerically and theoretically and thus we argue that our findings are crucial on the way to use NISQ computers in practical applications. Hence Paper B ties nicely into Paper A, since we saw areas where the proposed algorithm struggled, and it might be that utilizing appropriate error mitigation techniques could enhance the performance in these ranges.

- **Contribute to algorithmic agency ML approaches that learn how to exploit similarities in quantum physical experiments.** Exploiting similarities in physical experiments can possibly lower the computational resources required. This might also be the case of quantum spin Hamiltonians. Using a combination of active learning (AL) and Bayesian optimization (BO), Paper C (Chapter 6) proposes a novel active algorithmic agent which go beyond hyperparameter tuning of quantum neural networks for specific spin models as it also finds the next Hamiltonians to try out using uncertainty sampling. This contribution thus exploits the fact that some quantum physical experiments are similar and that hyperparameters (and in some cases parameters for warm start) can be transferred between experiments. We call the approach the *actively learned bayesian optimized* (ALBO) protocol. An important aspect for the ALBO is the uncertainty estimates provided by the surrogate model. Together with the mean function, they decide how to query new Hamiltonians as well as find the best set of hyperparamters with BO. Up until Paper D (Chapter 7), it was generally argued that calibrated uncertainties leads to better BO performance. However, as we show, this general statement might be misleading. In particular, we show that re-calibration uncertainties might be difficult to combine with BO given the small sample sizes often used in BO.

This finding is crucial for practitioners, as they should consider how to spend a potential limited budget of experiments.

We started off developing a VQA exploiting the noise in quantum circuits in order to perform sampling tasks, challenging for classical computers. However, there were certain instances of non-trivial patterns in the algorithms ability to prepare thermal states with high fidelity and thus sparked two new research directions. The first direction was getting to know the noise accumulation better and how this is distributed for some popular QML ansätze which resulted in coming up with unifying metrics ultimately tying into better coping with quantum error mitigation techniques. The second research direction was trying to learn how to design one's ansatz not only via Bayesian optimization, but also by exploiting what can already be obtained from previous similar experiments; the used example being groundstate estimation for popular spin models. By using a classical meta-learner, one learn from past experiences. This second research direction ultimately sparked the fourth and final research project concerning the role of the uncertainties in such meta-learners. Whereas the machine learning research community often argued that calibrated uncertainties aids the optimization process, little to no work was done proving focusing on this specific problem and getting rigorous results. With our fourth paper, we show that the uncertainty plays big role, but whether these uncertainties are calibrated does not seem to be crucial; only that they are in a range which allows for a useful balance of exploration/exploitation. However, going to the extend of re-calibration during Bayesian Optimization might be unnecessary in many instances due to the oftentimes small sample sizes used. The four contributions thus ties nicely together, and there is at least some overall connection of the work laid out.

We conclude this Ph.D. thesis by mentioning some potential interesting future paths along the lines of the contributed research for the QML research community. As companies are building larger NISQ devices, for example IBM planning on reaching over 4000 qubits in 2025 [43], experimental QML could potentially really start to take of in the coming years. Already now on the experimental side, there has been multiple interesting implementations of variational quantum algorithms [179, 286]. The findings of Paper A and B could be interesting to turn into actual physical experiments. Another interesting research alley is that of expanding on quantum error mitigation techniques and how this ties into the experimental side of things to use in practical applications as well as pave the way into quantum error correction. Third, there potentially exists many interesting options within leveraging more classical computational resources to empower noisy quantum experiments, just as we saw in Paper C. Specifically, we saw how combining AL an BO could accelerate hyperparameter tuning and thus convergence of VQE protocols

and one could imagine that more advanced ideas from state of the art machine learning could yield even more impressive results in a broader collection of problems. Outside the scientific contributions in this thesis exists several promising ideas; especially certain types of sequence learning [171] and exploiting symmetries to create geometrically motivated [287] ansätze.

By continuing to invest in projects such as this one we get closer to answering key questions that determines when and how we expect practical applicability of quantum machine learning.

Bibliography

[1] **J. Foldager**, A. Pesah, and L.K. Hansen. Noise-assisted variational quantum thermalization. *Scientific reports*, 12(1):1–11, 2022.

[2] **J. Foldager** and B. Koczor. Can shallow quantum circuits scramble local noise into global white noise? *arXiv preprint arXiv:2302.00881*, 2023.

[3] **J. Foldager**. Actively learning quantum machine learning architectures from related problems. 2023.

[4] **J. Foldager**, M. Jordahn, L.K. Hansen, and M.R. Andersen. On the role of model uncertainties in bayesian optimization. *arXiv preprint arXiv:2301.05983*, 2023.

[5] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

[6] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[9] Robert Dale. Gpt-3: What is it good for? *Natural Language Engineering*, 27(1):113–118, 2021.

[10] Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:681–694, 2020.

[11] Brady D Lund and Ting Wang. Chatting about chatgpt: how may ai and gpt impact academia and libraries? *Library Hi Tech News*, 2023.

[12] K Roose. The brilliance and weirdness of chatgpt. *The New York Times*, 2022.

[13] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S Kottmann, Tim Menke, and Alan Aspuru-Guzik. Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics*, 94(1): 015004, 2022.

[14] Jonathan Wei Zhong Lau, Kian Hwee Lim, Harshank Shrotriya, and Leong Chuan Kwek. Nisq computing: where are we and where do we go? *AAPPS Bulletin*, 32(1):27, 2022.

[15] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

[16] Matthias Möller and Cornelis Vuik. On the impact of quantum computing technology on future developments in high-performance scientific computing. *Ethics and information technology*, 19: 253–269, 2017.

[17] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[18] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[19] Xin Zhou and Xiaofei Tang. Research and implementation of rsa algorithm for encryption and decryption. In *Proceedings of 2011 6th international forum on strategic technology*, volume 2, pages 1118–1121. IEEE, 2011.

[20] Carl Pomerance. A tale of two sieves. *Notices of the American Mathematical Society*, 43(12): 1473–1485, 1996.

[21] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.

[22] Richard P Feynman. Plenty of room at the bottom. In *APS annual meeting*, 1959.

[23] Paul Benioff. The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. *Journal of statistical physics*, 22: 563–591, 1980.

[24] John Preskill. Lecture notes for physics 229: Quantum information and computation. *California Institute of Technology*, 16(1):1–8, 1998.

[25] David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818): 97–117, 1985.

[26] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907): 553–558, 1992.

[27] Richard Cleve, Artur Ekert, Chiara Macchiavello, and Michele Mosca. Quantum algorithms revisited. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):339–354, 1998.

[28] Mark Hillery. Coherence as a resource in decision problems: The deutsch-jozsa algorithm and a variation. *Physical Review A*, 93(1):012111, 2016.

[29] Poornima Aradyamath, NM Naghabhushana, and Rohitha Ujjinimatad. Quantum computing concepts with deutsch jozsa algorithm. *JOIV: International Journal on Informatics Visualization*, 3 (1):59–68, 2019.

[30] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

[31] Tanay Roy, Liang Jiang, and David I Schuster. Deterministic grover search with a restricted oracle. *Physical Review Research*, 4(2):L022013, 2022.

[32] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996.

[33] Richard P Feynman. Simulating physics with computers. In *Feynman and computation*, pages 133–153. CRC Press, 2018.

[34] Microsoft Quantum. Stephen Jordan. Quantum algorithm zoo. `https://quantumalgorithmzoo.org/`. [Online; accessed 19-June-2021].

[35] JM Raimond and S Haroche. Quantum computing: dream or nightmare. *31st Rencontres de Moriond: Dark Matter and Cosmology, Quantum Measurements and Experimental Gravitation*, pages 341–346, 1996.

[36] William G Unruh. Maintaining coherence in quantum computers. *Physical Review A*, 51(2):992, 1995.

[37] Peter W Shor. Scheme for reducing decoherence in quantum computer memory. *Physical review A*, 52(4):R2493, 1995.

[38] Andrew M Steane. Error correcting codes in quantum theory. *Physical Review Letters*, 77(5):793, 1996.

[39] Emanuel Knill. Quantum computing with realistically noisy devices. *Nature*, 434(7029):39–44, 2005.

[40] Emanuel Knill, Raymond Laflamme, and Wojciech Zurek. Threshold accuracy for quantum computation. *arXiv preprint quant-ph/9610011*, 1996.

[41] A Yu Kitaev. Quantum error correction with imperfect gates. *Quantum communication, computing, and measurement*, pages 181–188, 1997.

[42] Dorit Aharonov and Michael Ben-Or. Fault-tolerant quantum computation with constant error. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 176–188, 1997.

[43] The ibm quantum development roadmap. `https://www.ibm.com/quantum/roadmap`. Accessed: 08-03-2023.

[44] Lee Gomes. Quantum computing: Both here and not here. *IEEE Spectrum*, 55(4):42–47, 2018.

[45] Ben Sussman, Paul Corkum, Alexandre Blais, David Cory, and Andrea Damascelli. Quantum canada. *Quantum Science and Technology*, 4(2):020503, 2019.

[46] Quantum motion adds £ 42 million to accelerated funding of quantum computing startups. https://www.forbes.com/sites/gilpress/2023/02/20/quantum-motion-adds-42-million-to-accelerated-funding-of-quantum-computing-startups/. Accessed: 09-03-2023.

[47] Craig Gidney and Martin Ekerå. How to factor 2048 bit rsa integers in 8 hours using 20 million noisy qubits. *Quantum*, 5:433, 2021.

[48] Jacob D Biamonte, Pavel Dorozhkin, and Igor Zacharov. Keep quantum computing global and open, 2019.

[49] Jean-François Bobier, Matt Langione, Edward Tao, and Antoine Gourevitch. What happens when âifâturns to âwhenâin quantum computing. *Boston Consulting Group*, 2021.

[50] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[51] Francis Weston Sears, Mark Waldo Zemansky, Hugh David Young, and Roger A Freedman. *Sears and Zemansky's University Physics: With Modern Physics: Technology Update*. Pearson, 2014.

[52] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[53] Barry Gower. *Scientific method: A historical and philosophical introduction*. Routledge, 2012.

[54] Neil deGrasse Tyson. *Astrophysics for People in a Hurry*. WW Norton & Company, 2017.

[55] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[56] Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. *Explainable AI: interpreting, explaining and visualizing deep learning*, volume 11700. Springer Nature, 2019.

[57] Nils Thuerey, Philipp Holl, Maximilian Mueller, Patrick Schnell, Felix Trost, and Kiwon Um. Physics-based deep learning. *arXiv preprint arXiv:2109.05237*, 2021.

[58] Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning. *arXiv preprint arXiv:1710.05468*, 2017.

[59] Lei Wu, Zhanxing Zhu, et al. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239*, 2017.

[60] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4):045002, 2019.

[61] M Cerezo, Guillaume Verdon, Hsin-Yuan Huang, Lukasz Cincio, and Patrick J Coles. Challenges and opportunities in quantum machine learning. *Nature Computational Science*, 2(9):567–576, 2022.

[62] George EP Box. Science and statistics. *Journal of the American Statistical Association*, 71(356): 791–799, 1976.

[63] Jan Faye. Copenhagen interpretation of quantum mechanics. 2002.

[64] Allan Adams. Lecture 1: Introduction to superposition. In *Quantum Physics I—MIT Course No. 8.04*. Cambridge MA, 2013. URL `https://ocw.mit.edu/courses/8-04-quantum-physics-i-spring-2013/`. MIT OpenCourseWare.

[65] Allan Adams. Lecture 2: Experimental facts of life. In *Quantum Physics I—MIT Course No. 8.04*. Cambridge MA, 2013. URL `https://ocw.mit.edu/courses/8-04-quantum-physics-i-spring-2013/`. MIT OpenCourseWare.

[66] Leonard Susskind and Art Friedman. *Quantum mechanics: the theoretical minimum*. Basic Books, 2014.

[67] Walther Gerlach and Otto Stern. Der experimentelle nachweis des magnetischen moments des silberatoms. *Zeitschrift für Physik*, 8(1):110–111, 1922.

[68] R_P_ Feynman, RB Leighton, and M Sands. The feynman lectures on physics addison-wesley. *Reading, Ma*, 1:1–9, 1963.

[69] John Wertz. *Electron spin resonance: elementary theory and practical applications*. Springer Science & Business Media, 2012.

[70] Sternâgerlach experiment. `https://en.wikipedia.org/wiki/Stern-Gerlach_experiment`. Accessed: 30-03-2023.

[71] Howard Percy Robertson. The uncertainty principle. *Physical Review*, 34(1):163, 1929.

[72] Art Hobson. Electrons as field quanta: A better way to teach quantum physics in introductory general physics courses. *American Journal of Physics*, 73(7):630–634, 2005.

[73] Louis De Broglie. Waves and quanta. *Nature*, 112(2815):540–540, 1923.

[74] Alhun Aydin. Quantum shape effects. *arXiv preprint arXiv:2102.04332*, 2021.

[75] Jean Dalibard and Sylvain Gigan. A nobel prize for alain aspect, john clauser and anton zeilinger. *Photoniques*, 116:23–25, 2022.

[76] Alain Aspect, Jean Dalibard, and Gérard Roger. Experimental test of bell's inequalities using time-varying analyzers. *Physical review letters*, 49(25):1804, 1982.

[77] Yuchen Wang, Zixuan Hu, Barry C Sanders, and Sabre Kais. Qudits and high-dimensional quantum computing. *Frontiers in Physics*, 8:589504, 2020.

[78] David E Bernal, Akshay Ajagekar, Stuart M Harwood, Spencer T Stober, Dimitar Trenev, and Fengqi You. Perspectives of quantum computing for chemical engineering. *AIChE Journal*, 68 (6):e17651, 2022.

[79] Xiu Gu, Anton Frisk Kockum, Adam Miranowicz, Yu-xi Liu, and Franco Nori. Microwave photonics with superconducting quantum circuits. *Physics Reports*, 718:1–102, 2017.

[80] M-H Yung, Jorge Casanova, Antonio Mezzacapo, Jarrod Mcclean, Lucas Lamata, Alan Aspuru-Guzik, and Enrique Solano. From transistor to trapped-ion computers for quantum chemistry. *Scientific reports*, 4(1):3589, 2014.

[81] Norbert M Linke, Dmitri Maslov, Martin Roetteler, Shantanu Debnath, Caroline Figgatt, Kevin A Landsman, Kenneth Wright, and Christopher Monroe. Experimental comparison of two quantum

computing architectures. *Proceedings of the National Academy of Sciences*, 114(13):3305–3310, 2017.

[82] Ryan LaRose. Overview and comparison of gate level quantum software platforms. *Quantum*, 3: 130, 2019.

[83] A Yu Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191, 1997.

[84] Bin Yan and Nikolai A Sinitsyn. An adiabatic oracle for grover's algorithm. *arXiv preprint arXiv:2207.05665*, 2022.

[85] Elham Kashefi, Adrian Kent, Vlatko Vedral, and Konrad Banaszek. Comparison of quantum oracles. *Physical Review A*, 65(5):050304, 2002.

[86] Anirudh Goyal and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A*, 478(2266):20210068, 2022.

[87] Daniel A Roberts, Sho Yaida, and Boris Hanin. *The principles of deep learning theory*. Cambridge University Press Cambridge, MA, USA, 2022.

[88] Maria Schuld and Francesco Petruccione. *Supervised learning with quantum computers*, volume 17. Springer, 2018.

[89] Timo Felser, Marco Trenti, Lorenzo Sestini, Alessio Gianelle, Davide Zuliani, Donatella Lucchesi, and Simone Montangero. Quantum-inspired machine learning on high-energy physics data. *npj Quantum Information*, 7(1):111, 2021.

[90] Ewin Tang. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st annual ACM SIGACT symposium on theory of computing*, pages 217–228, 2019.

[91] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.

[92] Daniel K Park, Francesco Petruccione, and June-Koo Kevin Rhee. Circuit-based quantum random access memory for classical data. *Scientific reports*, 9(1):3949, 2019.

[93] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum algorithms for supervised and unsupervised machine learning. *arXiv preprint arXiv:1307.0411*, 2013.

[94] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631–633, 2014.

[95] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical review letters*, 113(13):130503, 2014.

[96] Zhikuan Zhao, Jack K Fitzsimons, Michael A Osborne, Stephen J Roberts, and Joseph F Fitzsimons. Quantum algorithms for training gaussian processes. *Physical Review A*, 100(1):012304, 2019.

[97] Peter Wittek. *Quantum machine learning: what quantum computing means to data mining*. Academic Press, 2014.

[98] Chris Ferrie. Associate Professor @ University of Technology Sydney. Hereâs why superposition and entanglement have nothing to do with understanding quantum computers. `https://csferrie.medium.com`, Jun. 20, 2021. [Online; accessed 19-June-2021].

[99] MI Dyakonov. Is fault-tolerant quantum computation really possible. *Future Trends in Microelectronics. Up the Nano Creek*, page 4, 2007.

[100] Dave Bacon and Wim VAn DAm. Recent progress in quantum algorithms. *Communications of the ACM*, 53(2):84–93, 2010.

[101] Scott Aaronson. Multilinear formulas and skepticism of quantum computing. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 118–127, 2004.

[102] Scott Aaronson. Read the fine print. *Nature Physics*, 11(4):291–293, 2015.

[103] Scott Aaronson. How much structure is needed for huge quantum speedups? *arXiv preprint arXiv:2209.06930*, 2022.

[104] Ryan Babbush, Jarrod R McClean, Michael Newman, Craig Gidney, Sergio Boixo, and Hartmut Neven. Focus beyond quadratic speedups for error-corrected quantum advantage. *PRX Quantum*, 2(1):010103, 2021.

[105] Earl T Campbell, Barbara M Terhal, and Christophe Vuillot. Roads towards fault-tolerant universal quantum computation. *Nature*, 549(7671):172–179, 2017.

[106] Maria Schuld and Nathan Killoran. Is quantum advantage the right goal for quantum machine learning? *Prx Quantum*, 3(3):030101, 2022.

[107] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[108] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L Oâbrien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5(1):1–7, 2014.

[109] Jack D Hidary and Jack D Hidary. Dirac notation. *Quantum Computing: An Applied Approach*, pages 377–381, 2021.

[110] David Wallace. How to prove the born rule. *Many worlds*, pages 227–263, 2010.

[111] William K Wootters. Alternatives to standard quantum theory ruled out, 2021.

[112] Sean R Eddy. Hidden markov models. *Current opinion in structural biology*, 6(3):361–365, 1996.

[113] Christopher J Wood, Jacob D Biamonte, and David G Cory. Tensor networks and graphical calculus for open quantum systems. *arXiv preprint arXiv:1111.6950*, 2011.

[114] Koenraad MR Audenaert and Stefan Scheel. On random unitary channels. *New Journal of Physics*, 10(2):023011, 2008.

[115] Bryce Seligman DeWitt and Neill Graham. *The many-worlds interpretation of quantum mechanics*, volume 61. Princeton University Press, 2015.

[116] Hugh Everett III. " relative state" formulation of quantum mechanics. *Reviews of modern physics*, 29(3):454, 1957.

[117] Sean Carroll. *Something deeply hidden: Quantum worlds and the emergence of spacetime*. Penguin, 2020.

[118] Stephen Barnett. *Quantum information*, volume 16. Oxford University Press, 2009.

[119] Pennylane qml.stronglyentanglinglayers. `https://docs.pennylane.ai/en/stable/code/api/pennylane.StronglyEntanglingLayers.html`. Accessed: 27-03-2023.

[120] Tameem Albash and Daniel A Lidar. Adiabatic quantum computation. *Reviews of Modern Physics*, 90(1):015002, 2018.

[121] Michael Freedman, Alexei Kitaev, Michael Larsen, and Zhenghan Wang. Topological quantum computation. *Bulletin of the American Mathematical Society*, 40(1):31–38, 2003.

[122] Noisy circuits. `https://pennylane.ai/qml/demos/tutorial_noisy_circuits.html`,. Accessed: 30-03-2023.

[123] Jonathan A Jones. Composite pulses in nmr quantum computation. *arXiv preprint arXiv:0906.4719*, 2009.

[124] Samson Wang, Enrico Fontana, Marco Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nature communications*, 12(1):6961, 2021.

[125] Joel J Wallman and Joseph Emerson. Noise tailoring for scalable quantum computation via randomized compiling. *Physical Review A*, 94(5):052325, 2016.

[126] Zhenyu Cai and Simon C Benjamin. Constructing smaller pauli twirling sets for arbitrary error channels. *Scientific reports*, 9(1):11281, 2019.

[127] Steven T Flammia and Joel J Wallman. Efficient estimation of pauli channels. *ACM Transactions on Quantum Computing*, 1(1):1–32, 2020.

[128] Google quantum ai software package "cirq" definition and implementation of a moment. `https://quantumai.google/cirq/build/circuits`. Accessed: 30-03-2023.

[129] Mark M Wilde. From classical to quantum shannon theory. *arXiv preprint arXiv:1106.1445*, 2011.

[130] Barry A Cipra. An introduction to the ising model. *The American Mathematical Monthly*, 94(10): 937–959, 1987.

[131] Richard P Feynman. The principle of least action in quantum mechanics. In *Feynman's ThesisâA New Approach To Quantum Theory*, pages 1–69. World Scientific, 2005.

[132] Pierre Pfeuty. The one-dimensional ising model with a transverse field. *ANNALS of Physics*, 57 (1):79–90, 1970.

[133] Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local hamiltonian problem. *Siam journal on computing*, 35(5):1070–1097, 2006.

[134] Max Born and Vladimir Fock. Beweis des adiabatensatzes. *Zeitschrift für Physik*, 51(3):165–180, 1928.

[135] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*, 2000.

[136] Hale F Trotter. On the product of semi-groups of operators. *Proceedings of the American Mathematical Society*, 10(4):545–551, 1959.

[137] Yin Sun, Jun-Yi Zhang, Mark S Byrd, and Lian-Ao Wu. Adiabatic quantum simulation using trotterization. *arXiv preprint arXiv:1805.11568*, 2018.

[138] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Daniel Preda. A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem. *Science*, 292(5516):472–475, 2001.

[139] D-wave systems. `https://www.dwavesys.com`. Accessed: 31-03-2023.

[140] Wim Van Dam, Michele Mosca, and Umesh Vazirani. How powerful is adiabatic quantum computation? In *Proceedings 42nd IEEE symposium on foundations of computer science*, pages 279–287. IEEE, 2001.

[141] Edwin T Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

[142] Pablo A Morales and Fernando E Rosas. A generalization of the maximum entropy principle for curved statistical manifolds. *arXiv preprint arXiv:2105.07953*, 2021.

[143] Stuart J Russell. *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.

[144] Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 44(1.2):206–226, 2000.

[145] Sang Gyu Kwak and Jong Hae Kim. Central limit theorem: the cornerstone of modern statistics. *Korean journal of anesthesiology*, 70(2):144–156, 2017.

[146] BD Craven and Sardar MN Islam. Ordinary least-squares regression. *The SAGE dictionary of quantitative management research*, pages 224–228, 2011.

[147] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[148] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

[149] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2015.

[150] Aryan Deshwal, Syrine Belakaria, and Janardhan Rao Doppa. Bayesian optimization over hybrid spaces. *arXiv preprint arXiv:2106.04682*, 2021.

[151] Ian Dewancker, Michael McCourt, and Scott Clark. Bayesian optimization for machine learning: A practical guidebook. *arXiv preprint arXiv:1612.04858*, 2016.

[152] Aaron Klein, Stefan Falkner, Simon Bartels, Philipp Hennig, and Frank Hutter. Fast bayesian optimization of machine learning hyperparameters on large datasets. In *Artificial intelligence and statistics*, pages 528–536. PMLR, 2017.

[153] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

[154] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[155] Jost Tobias Springenberg, Aaron Klein, Stefan Falkner, and Frank Hutter. Bayesian optimization with robust bayesian neural networks. *Advances in neural information processing systems*, 29: 4134–4142, 2016.

[156] Expected improvement for bayesian optimization: A derivation. `http://ash-aldujaili.github.io/blog/2018/02/01/ei/`. Accessed: 04-04-2023.

[157] Burr Settles. Active learning literature survey. 2009.

[158] Zoubin Ghahramani. Unsupervised learning. *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures*, pages 72–112, 2004.

[159] Geoffrey E Hinton. Boltzmann machine. *Scholarpedia*, 2(5):1668, 2007.

[160] Draw restricted boltzmann machines using tikz. `https://gist.github.com/stwind/999544e9002e0aa477653fddf95d4dc59`. Accessed: 28-03-2023.

[161] Nathan Killoran (Xanadu) Masoud Mohseni (Google Quantum AI) Peter Wittek. Patrick Huembeli (EPFL), Juan Miguel Arrazola (Xanadu). The physics of energy-based models. `https://towardsdatascience.com/the-physics-of-energy-based-models-1121122d0d9`, Jun. 30 2021. [Online; accessed 29-March-2023].

[162] Guillaume Verdon, Michael Broughton, Jarrod R McClean, Kevin J Sung, Ryan Babbush, Zhang Jiang, Hartmut Neven, and Masoud Mohseni. Learning to learn with quantum neural networks via classical neural networks. *arXiv preprint arXiv:1907.05415*, 2019.

[163] Guillaume Verdon, Trevor McCourt, Enxhell Luzhnica, Vikash Singh, Stefan Leichenauer, and Jack Hidary. Quantum graph neural networks. *arXiv preprint arXiv:1909.12264*, 2019.

[164] Michael Broughton, Guillaume Verdon, Trevor McCourt, Antonio J Martinez, Jae Hyeon Yoo, Sergei V Isakov, Philip Massey, Ramin Halavati, Murphy Yuezhen Niu, Alexander Zlokapa, et al. Tensorflow quantum: A software framework for quantum machine learning. *arXiv preprint arXiv:2003.02989*, 2020.

[165] Martin Larocca, Nathan Ju, Diego García-Martín, Patrick J Coles, and Marco Cerezo. Theory of overparametrization in quantum neural networks. *arXiv preprint arXiv:2109.11676*, 2021.

[166] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[167] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.

[168] EM Stoudenmire and Xavier Waintal. Grover's algorithm offers no quantum advantage. *arXiv preprint arXiv:2303.11317*, 2023.

[169] Eric R Anschuetz and Bobak T Kiani. Quantum variational algorithms are swamped with traps. *Nature Communications*, 13(1):7760, 2022.

[170] Amira Abbas, David Sutter, Christa Zoufal, Aurélien Lucchi, Alessio Figalli, and Stefan Woerner. The power of quantum neural networks. *Nature Computational Science*, 1(6):403–409, 2021.

[171] Eric R Anschuetz, Hong-Ye Hu, Jin-Long Huang, and Xun Gao. Interpretable quantum advantage in neural sequence learning. *arXiv preprint arXiv:2209.14353*, 2022.

[172] Gavin E Crooks. Performance of the quantum approximate optimization algorithm on the maximum cut problem. *arXiv preprint arXiv:1811.08419*, 2018.

[173] Seth Lloyd. Quantum approximate optimization is computationally universal. *arXiv preprint arXiv:1812.11075*, 2018.

[174] Jacob Biamonte. Universal variational quantum computation. *Physical Review A*, 103(3): L030401, 2021.

[175] Edward Farhi and Aram W Harrow. Quantum supremacy through the quantum approximate optimization algorithm. *arXiv preprint arXiv:1602.07674*, 2016.

[176] Javier Alcazar, Vicente Leyton-Ortega, and Alejandro Perdomo-Ortiz. Classical versus quantum models in machine learning: insights from a finance application. *Machine Learning: Science and Technology*, 1(3):035003, 2020.

[177] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, 2015.

[178] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016.

[179] Peter JJ OâMalley, Ryan Babbush, Ian D Kivlichan, Jonathan Romero, Jarrod R McClean, Rami Barends, Julian Kelly, Pedram Roushan, Andrew Tranter, Nan Ding, et al. Scalable quantum simulation of molecular energies. *Physical Review X*, 6(3):031007, 2016.

[180] Gene H Golub and Henk A Van der Vorst. Eigenvalue computation in the 20th century. *Journal of Computational and Applied Mathematics*, 123(1-2):35–65, 2000.

[181] Daniel S Abrams and Seth Lloyd. Simulation of many-body fermi systems on a universal quantum computer. *Physical Review Letters*, 79(13):2586, 1997.

[182] Subhash C Kak. Quantum neural computing. *Advances in imaging and electron physics*, 94: 259–313, 1995.

[183] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. The quest for a quantum neural network. *Quantum Information Processing*, 13:2567–2586, 2014.

[184] Yudong Cao, Gian Giacomo Guerreschi, and Alán Aspuru-Guzik. Quantum neuron: an elementary building block for machine learning on quantum computers. *arXiv preprint arXiv:1711.11240*, 2017.

[185] Erik Torrontegui and Juan José García-Ripoll. Unitary quantum perceptron as efficient universal approximator (a). *EPL (Europhysics Letters)*, 125(3):30004, 2019.

[186] Francesco Tacchino, Panagiotis Barkoutsos, Chiara Macchiavello, Ivano Tavernelli, Dario Gerace, and Daniele Bajoni. Quantum implementation of an artificial feed-forward neural network. *Quantum Science and Technology*, 5(4):044010, 2020.

[187] Iris Cong, Soonwon Choi, and Mikhail D Lukin. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, 2019.

[188] Kerstin Beer, Dmytro Bondarenko, Terry Farrelly, Tobias J Osborne, Robert Salzmann, Daniel Scheiermann, and Ramona Wolf. Training deep quantum neural networks. *Nature communications*, 11(1):1–6, 2020.

[189] Zoë Holmes, Nolan Coble, Andrew T Sornborger, and Yiğit Subaşı. On nonlinear transformations in quantum computation. *arXiv preprint arXiv:2112.12307*, 2021.

[190] Kwok Ho Wan, Oscar Dahlsten, Hlér Kristjánsson, Robert Gardner, and MS Kim. Quantum generalisation of feedforward neural networks. *npj Quantum information*, 3(1):36, 2017.

[191] Michael Lubasch, Jaewoo Joo, Pierre Moinier, Martin Kiffner, and Dieter Jaksch. Variational quantum algorithms for nonlinear problems. *Physical Review A*, 101(1):010301, 2020.

[192] Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Physical review letters*, 122(4):040504, 2019.

[193] Maria Schuld. Supervised quantum machine learning models are kernel methods. *arXiv preprint arXiv:2101.11020*, 2021.

[194] Thomas Hubregtsen, David Wierichs, Elies Gil-Fuster, Peter-Jan HS Derks, Paul K Faehrmann, and Johannes Jakob Meyer. Training quantum embedding kernels on near-term quantum computers. *Physical Review A*, 106(4):042431, 2022.

[195] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4(4):043001, 2019.

[196] Marcello Benedetti, Delfina Garcia-Pintos, Oscar Perdomo, Vicente Leyton-Ortega, Yunseong Nam, and Alejandro Perdomo-Ortiz. A generative modeling approach for benchmarking and training shallow quantum circuits. *npj Quantum Information*, 5(1):1–9, 2019.

[197] Vicente Leyton-Ortega, Alejandro Perdomo-Ortiz, and Oscar Perdomo. Robust implementation of generative modeling with parametrized quantum circuits. *Quantum Machine Intelligence*, 3(1): 17, 2021.

[198] Masaya Kohda, Ryosuke Imai, Keita Kanno, Kosuke Mitarai, Wataru Mizukami, and Yuya O Nakagawa. Quantum expectation-value estimation by computational basis sampling. *Physical Review Research*, 4(3):033173, 2022.

[199] Oscar Higgott, Daochen Wang, and Stephen Brierley. Variational quantum computation of excited states. *Quantum*, 3:156, 2019.

[200] Jarrod R McClean, Mollie E Kimchi-Schwartz, Jonathan Carter, and Wibe A De Jong. Hybrid quantum-classical hierarchy for mitigation of decoherence and determination of excited states. *Physical Review A*, 95(4):042308, 2017.

[201] Ryan LaRose, Arkin Tikku, Étude OâNeel-Judy, Lukasz Cincio, and Patrick J Coles. Variational quantum state diagonalization. *npj Quantum Information*, 5(1):1–10, 2019.

[202] M Cerezo, Kunal Sharma, Andrew Arrasmith, and Patrick J Coles. Variational quantum state eigensolver. *arXiv preprint arXiv:2004.01372*, 2020.

[203] Max Hunter Gordon, Marco Cerezo, Lukasz Cincio, and Patrick J Coles. Covariance matrix preparation for quantum principal component analysis. *PRX Quantum*, 3(3):030334, 2022.

[204] Xiaosi Xu, Jinzhao Sun, Suguru Endo, Ying Li, Simon C Benjamin, and Xiao Yuan. Variational algorithms for linear algebra. *Science Bulletin*, 66(21):2181–2188, 2021.

[205] A Yu Kitaev. Quantum measurements and the abelian stabilizer problem. *arXiv preprint quant-ph/9511026*, 1995.

[206] Alexei Yu Kitaev, Alexander Shen, Mikhail N Vyalyi, and Mikhail N Vyalyi. *Classical and quantum computation*. Number 47. American Mathematical Soc., 2002.

[207] Daniel S Abrams and Seth Lloyd. Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors. *Physical Review Letters*, 83(24):5162, 1999.

[208] Boaz Barak. Work with what youâve got. *Nature Physics*, 17(3):295–296, 2021.

[209] Yi-Kai Liu, Matthias Christandl, and Frank Verstraete. Quantum computational complexity of the n-representability problem: Qma complete. *Physical review letters*, 98(11):110503, 2007.

[210] Adam D Bookatz. Qma-complete problems. *arXiv preprint arXiv:1212.6312*, 2012.

[211] Gerardo Ortiz, James E Gubernatis, Emanuel Knill, and Raymond Laflamme. Quantum algorithms for fermionic simulations. *Physical Review A*, 64(2):022319, 2001.

[212] Grzegorz Swirszcz, Wojciech Marian Czarnecki, and Razvan Pascanu. Local minima in training of neural networks. *arXiv preprint arXiv:1611.06310*, 2016.

[213] Sharu Theresa Jose and Osvaldo Simeone. Free energy minimization: A unified framework for modelling, inference, learning, and optimization. *arXiv preprint arXiv:2011.14963*, 2020.

[214] Patrick Huembeli, Juan Miguel Arrazola, Nathan Killoran, Masoud Mohseni, and Peter Wittek. The physics of energy-based models. *Quantum Machine Intelligence*, 4(1):1, 2022.

[215] John Preskill. Quantum information chapter 10. quantum shannon theory. *Institute for Quantum Information and Matter California Institute of Technology*, 2016.

[216] Guang Hao Low and Isaac L Chuang. Hamiltonian simulation by qubitization. *Quantum*, 3:163, 2019.

[217] Xiao Yuan, Suguru Endo, Qi Zhao, Ying Li, and Simon C Benjamin. Theory of variational quantum simulation. *Quantum*, 3:191, 2019.

[218] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3):032331, 2019.

[219] Ryan Sweke, Frederik Wilde, Johannes Meyer, Maria Schuld, Paul K Fährmann, Barthélémy Meynard-Piganeau, and Jens Eisert. Stochastic gradient descent for hybrid quantum-classical optimization. *Quantum*, 4:314, 2020.

[220] Roeland Wiersema, Dylan Lewis, David Wierichs, Juan Carrasquilla, and Nathan Killoran. Here comes the su(n): multivariate quantum gates and gradients. *arXiv preprint arXiv:2303.11355*, 2023.

[221] Aram W Harrow and John C Napp. Low-depth gradient measurements can improve convergence in variational hybrid quantum-classical algorithms. *Physical Review Letters*, 126(14):140502, 2021.

[222] Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2): 251–276, 1998.

[223] Yudi Pawitan. *In all likelihood: statistical modelling and inference using likelihood.* Oxford University Press, 2001.

[224] Johannes Jakob Meyer. Fisher information in noisy intermediate-scale quantum applications. *Quantum*, 5:539, 2021.

[225] Suguru Endo, Zhenyu Cai, Simon C Benjamin, and Xiao Yuan. Hybrid quantum-classical algorithms and quantum error mitigation. *Journal of the Physical Society of Japan*, 90(3):032001, 2021.

[226] Sam McArdle, Tyson Jones, Suguru Endo, Ying Li, Simon C Benjamin, and Xiao Yuan. Variational ansatz-based quantum simulation of imaginary time evolution. *npj Quantum Information*, 5(1):75, 2019.

[227] Naoki Yamamoto. On the natural gradient for variational quantum eigensolver. *arXiv preprint arXiv:1909.05074*, 2019.

[228] James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. Quantum natural gradient. *Quantum*, 4:269, 2020.

[229] Bálint Koczor and Simon C Benjamin. Quantum natural gradient generalised to non-unitary circuits. *arXiv preprint arXiv:1912.08660*, 2019.

[230] Jacob L Beckey, M Cerezo, Akira Sone, and Patrick J Coles. Variational quantum algorithm for estimating the quantum fisher information. *arXiv preprint arXiv:2010.10488*, 2020.

[231] Bálint Koczor and Simon C Benjamin. Quantum analytic descent. *arXiv preprint arXiv:2008.13774*, 2020.

[232] Gregory Boyd and Bálint Koczor. Training variational quantum circuits with covar: covariance root finding with classical shadows. *Physical Review X*, 12(4):041022, 2022.

[233] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[234] David Wierichs, Christian Gogolin, and Michael Kastoryano. Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer. *Physical Review Research*, 2(4): 043246, 2020.

[235] Peter D Johnson, Jonathan Romero, Jonathan Olson, Yudong Cao, and Alán Aspuru-Guzik. Qvector: an algorithm for device-tailored quantum error correction. *arXiv preprint arXiv:1711.02249*, 2017.

[236] Sukin Sim, Peter D Johnson, and Alán Aspuru-Guzik. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, 2(12):1900070, 2019.

[237] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *nature*, 549(7671):242–246, 2017.

[238] Dave Wecker, Matthew B Hastings, and Matthias Troyer. Progress towards practical quantum variational algorithms. *Physical Review A*, 92(4):042303, 2015.

[239] Harper R Grimsley, Sophia E Economou, Edwin Barnes, and Nicholas J Mayhall. An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nature communications*, 10(1):3007, 2019.

[240] Leo Zhou, Sheng-Tao Wang, Soonwon Choi, Hannes Pichler, and Mikhail D Lukin. Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. *Physical Review X*, 10(2):021067, 2020.

[241] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature communications*, 9(1): 1–6, 2018.

[242] Zoë Holmes, Kunal Sharma, Marco Cerezo, and Patrick J Coles. Connecting ansatz expressibility to gradient magnitudes and barren plateaus. *PRX Quantum*, 3(1):010313, 2022.

[243] Laura Gentini, Alessandro Cuccoli, Stefano Pirandola, Paola Verrucchi, and Leonardo Banchi. Noise-resilient variational hybrid quantum-classical optimization. *Physical Review A*, 102(5): 052414, 2020.

[244] Arthur Pesah, Marco Cerezo, Samson Wang, Tyler Volkoff, Andrew T Sornborger, and Patrick J

Coles. Absence of barren plateaus in quantum convolutional neural networks. *Physical Review X*, 11(4):041011, 2021.

[245] Tyler Volkoff and Patrick J Coles. Large gradients via correlation in random parameterized quantum circuits. *Quantum Science and Technology*, 6(2):025008, 2021.

[246] Lennart Bittel and Martin Kliesch. Training variational quantum algorithms is np-hard. *Physical review letters*, 127(12):120502, 2021.

[247] Xuchen You and Xiaodi Wu. Exponentially many local minima in quantum neural networks. In *International Conference on Machine Learning*, pages 12144–12155. PMLR, 2021.

[248] Marco Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J Coles. Cost-function-dependent barren plateaus in shallow quantum neural networks. *arXiv e-prints*, pages arXiv–2001, 2020.

[249] Martin Larocca, Frédéric Sauvage, Faris M Sbahi, Guillaume Verdon, Patrick J Coles, and Marco Cerezo. Group-invariant quantum machine learning. *PRX Quantum*, 3(3):030341, 2022.

[250] Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu, and Dacheng Tao. Expressive power of parametrized quantum circuits. *Physical Review Research*, 2(3):033125, 2020.

[251] Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physical Review A*, 103(3):032430, 2021.

[252] Michael KĂźhn, Sebastian Zanker, Peter Deglmann, Michael Marthaler, and Horst Weiß. Accuracy and resource estimations for quantum chemistry on a near-term quantum computer. *Journal of chemical theory and computation*, 15(9):4764–4780, 2019.

[253] Quantum computing for quantum chemistry: a brief perspective. `https://pennylane.ai/blog/2021/11/quantum-computing-for-quantum-chemistry-a-brief-perspective/`, . Accessed: 03-04-2023.

[254] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H Booth, et al. The variational quantum eigensolver: a review of methods and best practices. *Physics Reports*, 986:1–128, 2022.

[255] Douglas G Altman and J Martin Bland. Standard deviations and standard errors. *Bmj*, 331(7521): 903, 2005.

[256] Artur F Izmaylov, Tzu-Ching Yen, and Ilya G Ryabinkin. Revising the measurement process in the variational quantum eigensolver: is it possible to reduce the number of separately measured operators? *Chemical science*, 10(13):3746–3755, 2019.

[257] William J Huggins, Kianna Wan, Jarrod McClean, Thomas E OâBrien, Nathan Wiebe, and Ryan Babbush. Nearly optimal quantum algorithm for estimating multiple expectation values. *Physical Review Letters*, 129(24):240501, 2022.

[258] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, 2020.

[259] Thomas E O'Brien, Michael Streif, Nicholas C Rubin, Raffaele Santagati, Yuan Su, William J Huggins, Joshua J Goings, Nikolaj Moll, Elica Kyoseva, Matthias Degroote, et al. Efficient quantum computation of molecular forces and other energy gradients. *Physical Review Research*, 4(4): 043210, 2022.

[260] Vladyslav Verteletskyi, Tzu-Ching Yen, and Artur F Izmaylov. Measurement optimization in the variational quantum eigensolver using a minimum clique cover. *The Journal of chemical physics*, 152(12):124114, 2020.

[261] Artur F Izmaylov, Tzu-Ching Yen, Robert A Lang, and Vladyslav Verteletskyi. Unitary partitioning approach to the measurement problem in the variational quantum eigensolver method. *Journal of chemical theory and computation*, 16(1):190–195, 2019.

[262] Hamza Jnane, Brennan Undseth, Zhenyu Cai, Simon C Benjamin, and Bálint Koczor. Multicore quantum computing. *Physical Review Applied*, 18(4):044064, 2022.

[263] Junyu Liu, Minzhao Liu, Jin-Peng Liu, Ziyu Ye, Yuri Alexeev, Jens Eisert, and Liang Jiang.

Towards provably efficient quantum algorithms for large-scale machine-learning models. *arXiv preprint arXiv:2303.03428*, 2023.

[264] Hsin-Yuan Huang, Michael Broughton, Jordan Cotler, Sitan Chen, Jerry Li, Masoud Mohseni, Hartmut Neven, Ryan Babbush, Richard Kueng, John Preskill, et al. Quantum advantage in learning from experiments. *arXiv preprint arXiv:2112.00778*, 2021.

[265] Zhenyu Cai, Ryan Babbush, Simon C Benjamin, Suguru Endo, William J Huggins, Ying Li, Jarrod R McClean, and Thomas E O'Brien. Quantum error mitigation. *arXiv preprint arXiv:2210.00921*, 2022.

[266] Samson Wang, Piotr Czarnik, Andrew Arrasmith, Marco Cerezo, Lukasz Cincio, and Patrick J Coles. Can error mitigation improve trainability of noisy variational quantum algorithms? *arXiv preprint arXiv:2109.01051*, 2021.

[267] Bálint Koczor. The dominant eigenvector of a noisy quantum state. *New Journal of Physics*, 23 (12):123047, 2021.

[268] Ying Li and Simon C Benjamin. Efficient variational quantum simulator incorporating active error minimization. *Physical Review X*, 7(2):021050, 2017.

[269] Alexander M Dalzell, Nicholas Hunter-Jones, and Fernando GSL Brandão. Random quantum circuits transform local noise into global white noise. *arXiv preprint arXiv:2111.14907*, 2021.

[270] Joseph Vovrosh, Kiran E Khosla, Sean Greenaway, Christopher Self, Myungshik S Kim, and Johannes Knolle. Simple mitigation of global depolarizing errors in quantum simulations. *Physical Review E*, 104(3):035309, 2021.

[271] Kento Tsubouchi, Takahiro Sagawa, and Nobuyuki Yoshioka. Universal cost bound of quantum error mitigation based on quantum estimation theory. *arXiv preprint arXiv:2208.09385*, 2022.

[272] William J Huggins, Sam McArdle, Thomas E OâBrien, Joonho Lee, Nicholas C Rubin, Sergio Boixo, K Birgitta Whaley, Ryan Babbush, and Jarrod R McClean. Virtual distillation for quantum error mitigation. *Physical Review X*, 11(4):041036, 2021.

[273] Bálint Koczor. Exponential error suppression for near-term quantum devices. *Physical Review X*, 11(3):031057, 2021.

[274] Thomas E O'Brien, G Anselmetti, Fotios Gkritsis, VE Elfving, Stefano Polla, William J Huggins, Oumarou Oumarou, Kostyantyn Kechedzhi, Dmitry Abanin, Rajeev Acharya, et al. Purification-based quantum error mitigation of pair-correlated electron simulations. *arXiv preprint arXiv:2210.10799*, 2022.

[275] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner. Variational quantum boltzmann machines. *Quantum Machine Intelligence*, 3:1–15, 2021.

[276] Mohammad H Amin, Evgeny Andriyash, Jason Rolfe, Bohdan Kulchytskyy, and Roger Melko. Quantum boltzmann machine. *Physical Review X*, 8(2):021050, 2018.

[277] Guillaume Verdon, Michael Broughton, and Jacob Biamonte. A quantum algorithm to train neural networks using low-depth circuits. *arXiv preprint arXiv:1712.05304*, 2017.

[278] Fernando GSL Brandao and Krysta M Svore. Quantum speed-ups for solving semidefinite programs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 415–426. IEEE, 2017.

[279] Sören Becker, Marcel Ackermann, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Interpreting and explaining deep neural networks for classification of audio signals. *CoRR*, abs/1807.03418, 2018.

[280] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

[281] Armands Strikis, Dayue Qin, Yanzhu Chen, Simon C Benjamin, and Ying Li. Learning-based quantum error mitigation. *PRX Quantum*, 2(4):040330, 2021.

[282] Shachi Deshpande and Volodymyr Kuleshov. Calibration improves bayesian optimization. *arXiv preprint arXiv:2112.04620*, 2021.

[283] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning*, pages 2796–2804. PMLR, 2018.

[284] Ashish Nanda, Deepak Puthal, Saraju P Mohanty, and Uma Choppali. A computing perspective of quantum cryptography [energy and security]. *IEEE Consumer Electronics Magazine*, 7(6):57–59, 2018.

[285] Nick S Blunt, Joan Camps, Ophelia Crawford, Róbert Izsák, Sebastian Leontica, Arjun Mirani, Alexandra E Moylett, Sam A Scivier, Christoph SĂźnderhauf, Patrick Schopf, et al. Perspective on the current state-of-the-art of quantum computing for drug discovery applications. *Journal of Chemical Theory and Computation*, 18(12):7001–7023, 2022.

[286] Cornelius Hempel, Christine Maier, Jonathan Romero, Jarrod McClean, Thomas Monz, Heng Shen, Petar Jurcevic, Ben P Lanyon, Peter Love, Ryan Babbush, et al. Quantum chemistry calculations on a trapped-ion quantum simulator. *Physical Review X*, 8(3):031022, 2018.

[287] Quynh T Nguyen, Louis Schatzki, Paolo Braccia, Michael Ragone, Patrick J Coles, Frederic Sauvage, Martin Larocca, and M Cerezo. Theory for equivariant quantum neural networks. *arXiv preprint arXiv:2210.08566*, 2022.

Appendix A

Paper A (published version)

THE FOLLOWING PAGES CONTAIN A COPY OF THE OPEN ACCESS VERSION OF THE PAPER AND SUPPLEMENTARY MATERIAL PUBLISHED IN SCIENTIFIC REPORTS — NATURE

scientific reports

Check for updates

OPEN

Noise-assisted variational quantum thermalization

Jonathan Foldager[1✉], Arthur Pesah[2] & Lars Kai Hansen[1]

Preparing thermal states on a quantum computer can have a variety of applications, from simulating many-body quantum systems to training machine learning models. Variational circuits have been proposed for this task on near-term quantum computers, but several challenges remain, such as finding a scalable cost-function, avoiding the need of purification, and mitigating noise effects. We propose a new algorithm for thermal state preparation that tackles those three challenges by exploiting the noise of quantum circuits. We consider a variational architecture containing a depolarizing channel after each unitary layer, with the ability to directly control the level of noise. We derive a closed-form approximation for the free-energy of such circuit and use it as a cost function for our variational algorithm. By evaluating our method on a variety of Hamiltonians and system sizes, we find several systems for which the thermal state can be approximated with a high fidelity. However, we also show that the ability for our algorithm to learn the thermal state strongly depends on the temperature: while a high fidelity can be obtained for high and low temperatures, we identify a specific range for which the problem becomes more challenging. We hope that this first study on noise-assisted thermal state preparation will inspire future research on exploiting noise in variational algorithms.

Noise is often considered to be one of the strongest adversaries of practical quantum computation. Decoherence effects due to a noisy environment can create errors in the final output of a circuit, destroying the advantage of many quantum algorithms. In contrast, noise is also what underlies stochastic processes, and is therefore a crucial element in classical computing, solving tasks such as sampling and optimization. In quantum systems, noise has also been shown to be a useful resource in several applications: carefully engineered dissipative processes can lead to universal quantum computation[1], shot-noise in the measurement process can drive variational algorithms out of local minima[2,3], and amplitude-damping channels can significantly improve quantum autoencoders for mixed states[4]. We investigate in the present paper a novel way to exploit noise in near-term quantum devices, with the objective of studying a central task in quantum computing: thermal state preparation.

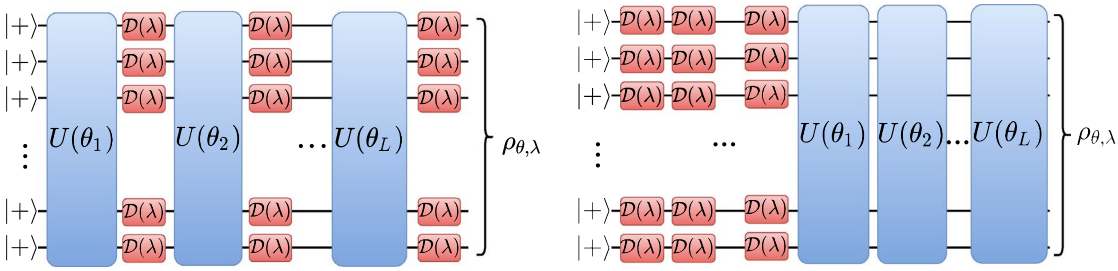
Placing a quantum system driven by a Hamiltonian H and weakly-coupled to a reservoir with an effective temperature $T = \frac{1}{\beta}$, the system will asymptotically reach a thermal equilibrium state, given by the quantum Gibbs distribution

$$\rho_\beta = \frac{1}{Z} e^{-\beta H} \tag{1}$$

where $Z = \mathrm{Tr}[e^{-\beta H}]$ is the partition function[5]. Efficiently preparing a thermal state on a quantum computer is a problem of broad practical importance, with applications ranging from quantum chemistry and many-body physics simulations in an open environment[6–8] to semi-definite programming[9,10] and quantum machine learning[11,12]. However, sampling from a general Gibbs distribution is a computationally hard task for classical computers, due to the complexity of calculating the partition function[13]. Most techniques rely on Monte-Carlo Markov Chain (MCMC) algorithms, which are often provably efficient only above a certain threshold temperature[14].

Many algorithms have been proposed to prepare the thermal state on a quantum computer. A growing body of work has suggested using variational algorithms to solve the task of thermal state preparation on Noisy Intermediate Scale Quantum (NISQ) devices. Since a unitary circuit acting on the zero-state cannot directly output a mixed state, most variational thermalization methods consist either in preparing a purification of the thermal state and tracing out the ancillary qubits at the end of the circuit[15–18], or in choosing an appropriate mixed state as input[19–21].

[1]Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800 Kongens Lyngby, Denmark. [2]Department of Physics and Astronomy, University College London, London WC1E 6BT, UK. ✉email: jonf@dtu.dk



(a) NAVQT ansatz.

(b) Approximation of the NAVQT ansatz.

Figure 1. Illustration of circuit components used in NAVQT. (**a**) General NAVQT ansatz: a sequence of unitary layers $U(\theta_i)$ followed by depolarizing gates $\mathcal{D}(\lambda)$ on each qubit. (**b**) Approximation used in the free-energy calculations.

One of the main challenges associated to those methods is to design an appropriate cost function to be minimized during the variational training loop. While the ground-state of a Hamiltonian can be prepared by minimizing the average energy of the state, the thermal state can be prepared by minimizing the *free energy* $F = H - TS$ of the state, where $S = -\mathrm{Tr}[\rho \log(\rho)]$ is its Von Neumann entropy. However, the Von Neumann entropy is not an observable and can often only be computed approximately[18,22]. A second problem is the need for additional qubits, which can be costly in near-term devices. Finally, none of those methods take into account the noise of the circuit, which can change the spectrum of the final state and affect the performance of the preparation algorithm[23].

In this paper, we propose a new method that we call *Noise-Assisted Variational Quantum Thermalizer* (NAVQT). Our algorithm assumes the ability to control the noise in the system down to some minimal noise level determined by the hardware. This type of control has been demonstrated in the context of error mitigation, where noise is increased in order to perform zero-noise extrapolation[24,25]. More precisely, we construct a variational circuit with a parametrized depolarizing channel after each layer of unitary gates, as illustrated in Fig. 1(a). To simplify the optimization process, we have only considered the case where all the depolarizing parameters take the same value. By varying both gate and noise parameters, we seek to minimize the free energy of the final state.

In order to compute the free energy (and its gradient), we derive an analytical expression for the entropy of a slightly different circuit: one where all the depolarizing gates have been displaced at the beginning of the circuit, as shown in Fig. 1(b). Using this approximation, we can compute the gradient of the free energy with respect to both the noise and the unitary parameters. While this might be a rough estimate of the actual gradient, we show that this approximate optimization problem exhibits similar performance as when minimizing the true free energy.

We then empirically investigate our algorithm on three different types of Hamiltonians: the Ising chain, with and without a transverse field, and the Heisenberg model. For each model, we consider both uniform coefficients and coefficients drawn from a standard normal distribution, and train our variational algorithm for several choices of hyperparameters (number of layers, learning rates, initialization, etc.). To study the performance of our approach, we extract the fidelity of the prepared state compared to the actual thermal state for a range of different temperatures.

Our results reveal different patterns. On the one hand, fidelities above 0.9 are reached for uniform Ising chains, with and without a transverse field, for all temperatures and system sizes up to 7 qubits. On the other hand, the performance tend to decrease with the system size and for specific ranges of temperatures, with fidelities that can get below 0.7 for some of the most complex systems tested in this work.

Our paper is organized as follows. We start by reviewing previous work on variational thermalization in "Related work" section. We then introduce NAVQT in "Noise-assisted variational quantum thermalization" section. We follow this up by a description of our experiments in "Methods" section, and present our results in "Results" section. Finally, we discuss our work and provide ideas for future studies in "Discussion" section.

Related work

Variational circuits have recently been proposed for thermal state preparation, due to the existence of a natural cost function for this task: the free energy. Using variational circuits to prepare a thermal state presents two challenges specific to this task: (1) finding an ansatz that can prepare mixed states, (2) finding a scalable optimization strategy.

Choice of the ansatz. A common approach to VQT consists in preparing a purification of the thermal state using a variational circuit that acts on $2N$ qubits—N system qubits and N ancilla/environment qubits—, and tracing the ancilla qubits out at the end of the circuit[15–18]. An example of purification often considered in the literature is the thermofield double (TFD) state[15,16]. For a Hamiltonian H and an inverse temperature β, it is given by

$$|\mathrm{TFD}\rangle = \frac{1}{\sqrt{Z}} \sum_n e^{-\beta E_n/2} |n\rangle_S \otimes |n\rangle_E \tag{2}$$

where the $\{E_n, |n\rangle\}_n$ are pairs of eigenvalue/eigenvector of H, and subscript S and E refers to the system and environment, respectively. For instance, Refs.[15,16] use a Quantum Approximate Optimization Ansatz (QAOA) ansatz acting on $2N$ qubits to prepare the TFD state of the transverse-field Ising model, the XY chain, and free fermions. One advantage of this approach is the ability to simulate the TFD, which can be interesting in in its own right, for instance for studying black holes[26]. The obvious disadvantage is that it requires twice as many qubits that the thermal state we want to simulate. A converse approach consists in starting with a mixed state ρ_0 and applying a unitary circuit ansatz on the N qubits of the system. The initial ρ_0 can either be fixed[19] or modified during the optimization process[20,21]. In Ref.[19], ρ_0 is the fixed thermal state of $H_I = \sum_{i=1}^N Z_i$, where Z_i is the Pauli Z operator applied to qubit i of the system. It can easily be prepared using the purification

$$\bigotimes_j \sqrt{2\cosh(\beta)} \sum_{b\in\{0,1\}^N} e^{(-1)^{1+b}\beta/2} |b\rangle_S |b\rangle_E. \tag{3}$$

However, since the spectrum does not change when we apply the unitary ansatz, having a static ρ_0 freezes the spectrum of the final state. Therefore, if the spectrum of the thermal state we want to approximate is far from the spectrum of ρ_0, this approach will fail. In Ref.[20], they use the thermal state $\rho_0(\boldsymbol{\varepsilon})$ of $H = \sum_{i=1}^n \varepsilon_i P_i$, where $P_i = \frac{1-Z_i}{2}$ as an initial state and $\boldsymbol{\varepsilon} = \{\varepsilon_1, \ldots, \varepsilon_n\}$ are parameters optimized during the training process. Finally, Ref.[21] proposes to use a unitary with stochastic parameters to prepare ρ_0. More precisely,

$$\rho_0(\boldsymbol{\theta}) = V(X_{\boldsymbol{\theta}})|0\rangle\langle 0|V(X_{\boldsymbol{\theta}})^\dagger \tag{4}$$

where $V(\boldsymbol{x})$ is a unitary ansatz and $X_{\boldsymbol{\theta}} \sim p_{\boldsymbol{\theta}}$ is a random vector with parametrized density $p_{\boldsymbol{\theta}}$. The density $p_{\boldsymbol{\theta}}$ can be given by a classical model, such as an energy-based model (e.g. restricted Boltzmann machine) or a normalizing flow, which will be trained to get a ρ_0 with a spectrum close to the thermal state of interest.

Optimization strategies. Once the ansatz has been fixed, the parameters within needs to be optimized. Two main approaches have been proposed in the literature: (1) explicitly minimizing the free energy, (2) using imaginary-time evolution. In the following, we describe both these methods.

Free energy methods. The thermal state is the density matrix that minimizes the free energy. Therefore, in the same way as VQE uses the energy as a cost function, any thermal state preparation method can use the free energy as its cost function[15,16,19,21]. However, one main difference with VQE is that the free energy cannot be easily estimated. Indeed, the Von Neumann entropy term, as a non-linear function of ρ, cannot be turned into an observable, and doing a full quantum state tomography would be very costly. Several methods have been proposed to solve this challenge:

- Computing several Renyi entropies $S_\alpha = \frac{1}{1-\alpha}\mathrm{Tr}[\rho^\alpha]$ (using multiple copies of ρ) and approximating the Von Neumann entropy with them[15,27].
- Computing the Von Neumann entropies locally on a small subsystem[15]
- Approximate the Von Neumann entropy by truncating its Taylor[18] or Fourier[22] decomposition.

In our work, the entropy term does not come from a purification procedure, but from the presence of depolarizing gates in the circuits. This led us to propose a different type of approximation that we will study in "Noise-assisted variational quantum thermalization".

Imaginary-time evolution. Thermal state preparation can be seen as the application of imaginary-time evolution during a time $\Delta t = i\beta/2$ on the maximally-mixed state $\rho_m = \frac{1}{d}\mathrm{I}$, using the decomposition

$$\rho_\beta = \left(\frac{1}{C}e^{-\beta H/2}\right)\left(\frac{1}{d}\mathrm{I}\right)\left(\frac{1}{C}e^{-\beta H/2}\right)$$

This imaginary-time evolution can be simulated using a variational circuit and a specific update rule[28,29]. In Ref.[17], the authors use a variational circuit $U(\boldsymbol{\theta})$ on $2N$ qubits, initialized such that

$$U(\boldsymbol{\theta}_0)|0\rangle^{\otimes 2N} \approx |\Phi^+\rangle$$

where Φ^+ is a maximally-entangled state. An imaginary-time update rule with a small learning rate τ will lead to a unitary $U(\boldsymbol{\theta}_0)$ such that:

$$U(\boldsymbol{\theta}_1)|0\rangle^{\otimes 2N} \approx \frac{1}{C}e^{-\tau H}|\Phi^+\rangle$$

Repeating it during $k = \frac{\beta}{2}$ steps will give the state

$$U(\boldsymbol{\theta}_k)|0\rangle^{\otimes 2N} \approx \frac{1}{C}e^{-\beta H/2}|\Phi^+\rangle$$

which will be the thermal state after tracing out the environment. In Ref.[30], the authors also use imaginary-time evolution to prepare the thermal state, but manage to reduce the number of qubits to N when the Hamiltonian is diagonal in the Z-basis. Finally, an ansatz-independent imaginary-time evolution method has been proposed for thermal state preparation[31,32].

In this work, we optimize the ansatz parameters using the free energy approach. Adapting imaginary-time evolution to a noisy ansatz could however be an interesting alternative, that we let for future work.

Noise-assisted variational quantum thermalization

We introduce here the *Noise-Assisted Variational Quantum Thermalizer* (NAVQT), a variational algorithm where depolarizing noise is used as the source of entropy for preparing the thermal state. We consider a noise model where each layer of unitary gates is followed by a one-qubit depolarizing channel

$$\mathcal{D}(\lambda)(\rho) = (1-\lambda)\rho + \lambda\frac{\mathrm{I}}{2}, \tag{5}$$

where I is the identity matrix. The channel is represented in Fig. 1. For the purpose of this work, we consider that we have the same noise value $\lambda \in [\lambda_{\min}, 1]$ everywhere in the circuit, where $\lambda_{\min}$ is the minimum noise reachable by the hardware. We note $\rho_{\boldsymbol{\theta},\lambda}$ the output of the noisy circuit with unitary parameters $\boldsymbol{\theta}$ and noise parameter λ, and want to find the optimal parameters $\{\boldsymbol{\theta}^*, \lambda^*\}$ such that $\rho_{\boldsymbol{\theta}^*,\lambda^*} \approx \rho_\beta$ where the latter is given by Eq. (1).

The thermal state ρ_β can be approximated by minimizing the free energy of the system, given by:

$$F(\boldsymbol{\theta}, \lambda) = E(\boldsymbol{\theta}, \boldsymbol{\lambda}) - \frac{1}{\beta}S(\boldsymbol{\theta}, \lambda) \tag{6}$$

where

$$E(\boldsymbol{\theta}, \lambda) = \mathrm{Tr}[H\rho_{\boldsymbol{\theta},\lambda}] \tag{7}$$

is the energy and

$$S(\boldsymbol{\theta}, \lambda) = -\mathrm{Tr}[\rho_{\boldsymbol{\theta},\lambda}\log(\rho_{\boldsymbol{\theta},\lambda})] \tag{8}$$

is the Von Neumann entropy of the state.

The energy term and its gradient are easy to evaluate: we can use the parameter shift-rule[33] to compute $\nabla_{\boldsymbol{\theta}}E(\boldsymbol{\theta}, \lambda)$, and the finite-difference method to calculate $\partial_\lambda E(\boldsymbol{\theta}, \lambda)$. The entropy term is much harder to evaluate as it is a non-linear function of the state. To approximate it, we consider the circuit where all the noise has been put at the beginning, as shown in Fig. 1(b). While the resulting free energy will not be equal to the free energy of our original circuit in general, they tend to follow similar trajectories when varying the noise level (see Supplementary Fig. S1). The new entropy does not depend on $\boldsymbol{\theta}$ and can be calculated analytically as if there were no unitary gates. For a circuit with N qubits and m layers, this approximate entropy $\widetilde{S}(\lambda)$ is given by

$$\begin{aligned}\widetilde{S}(\lambda) = -N\left((1-\lambda)^m + \frac{(1-(1-\lambda)^m)}{d}\right)\cdot \\ \ln\left((1-\lambda)^m + \frac{(1-(1-\lambda)^m)}{d}\right) \\ + \frac{(d-1)(1-(1-\lambda)^m)}{d}\ln\left(\frac{(1-(1-\lambda)^m)}{d}\right)\end{aligned} \tag{9}$$

where $d = 2^N$. The full derivation is given in the Supplementary material. Using this approximation, we get the following gradient-based update rule at each optimization step:

$$\boldsymbol{\theta}^{(n+1)} = \boldsymbol{\theta}^{(n)} - \eta_\theta \nabla_{\boldsymbol{\theta}}E(\boldsymbol{\theta}, \lambda) \tag{10}$$

$$\lambda^{(n+1)} = \lambda^{(n)} - \eta_\lambda\left(\nabla_\lambda E(\boldsymbol{\theta}, \lambda) - \frac{1}{\beta}\nabla_\lambda\widetilde{S}(\lambda)\right) \tag{11}$$

where η_θ and η_λ are the learning rates for $\boldsymbol{\theta}$ and λ, respectively.

Methods

In this section, we will briefly describe the basis of conducted experiments. All quantum circuit simulations are done in Cirq[34] and TensorFlow-Quantum[35].

Ansatz. For the unitary layers of our circuit, we chose an ansatz inspired by the Quantum Approximate Optimization Ansatz (QAOA) applied to the Ising chain Hamiltonian[36]. More precisely, if we define a problem Hamiltonian

$$H_P = -\sum_i Z_i Z_{i+1} - \sum_i Z_i \tag{12}$$

and a mixing Hamiltonian

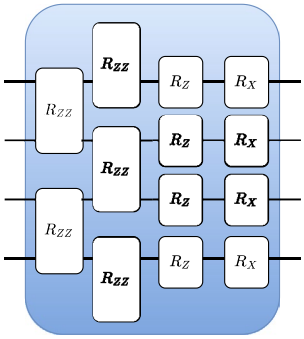


Figure 2. A layer of the unitary ansatz used in our experiments, inspired by QAOA for the 1D Ising model. R_Z and R_X are parametrized rotations around the corresponding axis, and $R_{ZZ} = e^{-i\theta Z_i Z_j}$.

$$H_M = -\sum_i X_i, \tag{13}$$

the QAOA ansatz with p layers is given by

$$U(\boldsymbol{\gamma}, \boldsymbol{\beta}) = e^{i\beta_p H_M} e^{i\gamma_p H_P} \dots e^{i\beta_1 H_M} e^{i\gamma_1 H_P} \tag{14}$$

This ansatz, whose explicit construction is represented in Fig. 2, has been well-studied in the context of ground-state preparation[37] and has been shown to be universal[38] in the limit $p \to \infty$. We test two different versions of this ansatz. In the first one, denoted *restricted QAOA*, gates of the same type from a given layer share the same parameters β_i and γ_i. In the second version, which we call *flexible QAOA*, every gate has its own parameter.

We ran some preliminary tests to verify that this unitary ansatz is at least able to express the ground-state of all the systems tested in our work, and found it to be the case when the number of layers is fixed at $\lceil \frac{N}{2} \rceil$. Hence the noisy ansatz should in principle be able to represent the correct thermal state for large β, by setting $\lambda = 0$ and fitting the unitary parameters corresponding to the ground-state. Moreover, NAVQT is also able to represent the maximally-mixed state, corresponding to a low β, by setting $\lambda = 1$. In Supplementary Figure S3, we provide some results for a varying number of layers $L \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$ at $\beta = 1$ for the three-qubit Heisenberg Hamiltonian with random coefficients, showing that the fidelity does not improve significantly compared to our heuristic number of layers. Hence we find evidence to rule out the number of ansatz layers as a limiting factor to achieve better performance. The ability of the ansatz to learn intermediate temperatures is an open question, that we tackle in our numerical analysis.

Hyperparameters. Since the choice of hyperparameters can have a substantial impact on the performance of variational circuits[37], we perform a grid-search to reduce the potential negative effects resulting from a single design choice. Hence we try all combinations in the search space defined by

- Restricted QAOA and flexible QAOA
- Initial noise level: $\lambda = \{10^{-8}, 0.001, 0.1\}$
- Unitary learning rate: $\eta_\theta = \{0.01, 0.4\}$
- Noise learning rate: $\eta_\lambda = \{0.0001, 0.1\}$
- Seeds for unitary parameters: $[0; 4]$.

We run our algorithm for $N \in [3; 7]$ qubits and for maximum 1000 iterations. To test the performance across temperatures, we take 10 different betas in the interval $\beta \in [10^{-3}; 10^2]$, namely $\{0.001, 0.1, 0.25, 0.5, 0.75, 1.0, 2.0, 5.0, 10.0, 100.0\}$. We initialize the unitary parameters by sampling from a uniform distribution in the interval $[0.0001, 0.05]$ as done in[37]. Finally, we extract the solution that gives the lowest (approximated) free energy among all the tested hyperparameters and initializations. We also include the same grid-search using finite-difference on the true free-energy in Supplementary Fig. S2.

Noisy circuit simulation. To simulate the noise in our circuit, we use the fact that depolarizing gates can also be written as[39]

$$\mathcal{D}(\lambda)(\rho) = \left(1 - \frac{3\lambda}{4}\right)\rho + \frac{\lambda}{4}(X\rho X + Y\rho Y + Z\rho Z) \tag{15}$$

which can be interpreted as applying a random Pauli error with probability $p = \frac{3\lambda}{4}$ and nothing with probability $p = 1 - \frac{3\lambda}{4}$. We can therefore simulate depolarizing gates as stochastic mixtures over unitary circuits containing errors. More precisely, if we sample K unitaries $U^{(k)}$, each being a combination of the unitary part of the ansatz and some random errors, we can extract the corresponding density matrix as:

$$\rho_{\text{out}} \approx \frac{1}{K}\sum_{k=1}^{K} U^{(k)}\rho_{in}\left(U^{(k)}\right)^{\dagger} \tag{16}$$

We found that taking a sample size of $K = 500N$ was sufficient to get stable gradients and reach the maximum entropy $S \leq \log 2^N$. However, we also found that K could be smaller, especially when β was large and hence the target entropy was low.

Performance metric. For each experiment, we report the fidelity

$$F(\rho_1, \rho_2) = \text{Tr}\left[\sqrt{\sqrt{\rho_1}\rho_2\sqrt{\rho_1}}\right] \tag{17}$$

between the thermal state and the output state of the trained circuit. Tracking the fidelity requires us to compute the true thermal state ρ_β for each Hamiltonian H and temperature β. In practice, taking the exponential of a matrix containing potentially large numerical values (e.g. when β is large) can result in numerical issues. To alleviate those issues, we calculate the thermal state density matrix ρ_β by taking the log on both sides of Eq. (1) and using the log-sum-exp trick[40]:

$$\begin{aligned}\log \rho_\beta &= \log e^{-\beta H} - \log \text{Tr}[e^{-\beta H}] \\ &= -\beta H - \log \sum_i e^{-\beta\lambda_i} \\ &= -\beta H - \left(-\beta c + \log \sum_i e^{-\beta(\lambda_i - c)}\right)\end{aligned} \tag{18}$$

where c is the largest eigenvalue of H.

Models. We evaluated our algorithm on three different models: the Ising chain, with and without a transverse field, and the Heisenberg model. For each model, we considered two cases: when the coefficients $J_i = h_i = 1$ for all i, denoted the *uniform* version, and when $J_i, h_i \sim \mathcal{N}(0, 1)$ for all i, denoted the *random* version. Between five seeds for the random version, we pick the Hamiltonian with the lowest spectral gap as this could be considered the hardest Hamiltonian. In the case for Hamiltonians with random coefficients, we normalized the set of all coefficients such that the vector containing all coefficients had unit length. See Supplementary Fig. S4 for a plot of the model energy scales.

Ising chain. The 1D Ising model, or Ising chain (IC), considers a set of spins on a chain such that all spins have exactly two coupled neighbors when considering $N > 2$. The Hamiltonian associated with such system is given by

$$H_{\text{IC}} = -\sum_i J_i Z_i Z_{i+1} - \sum_i h_i Z_i \tag{19}$$

where Z_i is the Pauli Z operator acting on qubit i.

Transverse field Ising chain. The transverse-field Ising chain (TFI) adds quantum effects to the previous model by including some non-diagonal terms in its Hamiltonian. It is defined as

$$H_{\text{TFI}} = -\sum_i J_i^Z Z_i Z_{i+1} - \sum_i h_i^Z Z_i - \sum_i h_i^X X_i \tag{20}$$

where X_i is the Pauli X operator acting on qubit i.

Heisenberg model. Finally, we consider the 1D Heisenberg model, whose Hamiltonian is given by

$$\begin{aligned}H_{\text{Heisenberg}} = &-\sum_i J_i^Z Z_i Z_{i+1} - \sum_i J_i^X X_i X_{i+1} \\ &- \sum_i J_i^Y Y_i Y_{i+1} - \sum_i h_i^X X_i\end{aligned} \tag{21}$$

The Heisenberg model is of fundamental importance in the study of quantum materials[41–44] and is therefore a standard benchmark for thermal state preparation methods[31,32,45].

Results

We first present the optimization curves for N = 4, at three different temperatures $\beta \in \{0.1, 0.5, 10\}$ in Fig. 3. We report the fidelity between the learned state and the thermal state as a function of the inverse temperature β for all the different models in Fig. 4. Finally, we also report the final noise level λ as a function of β for all models in Fig. 5. We can notice a few phenomena from those figures:

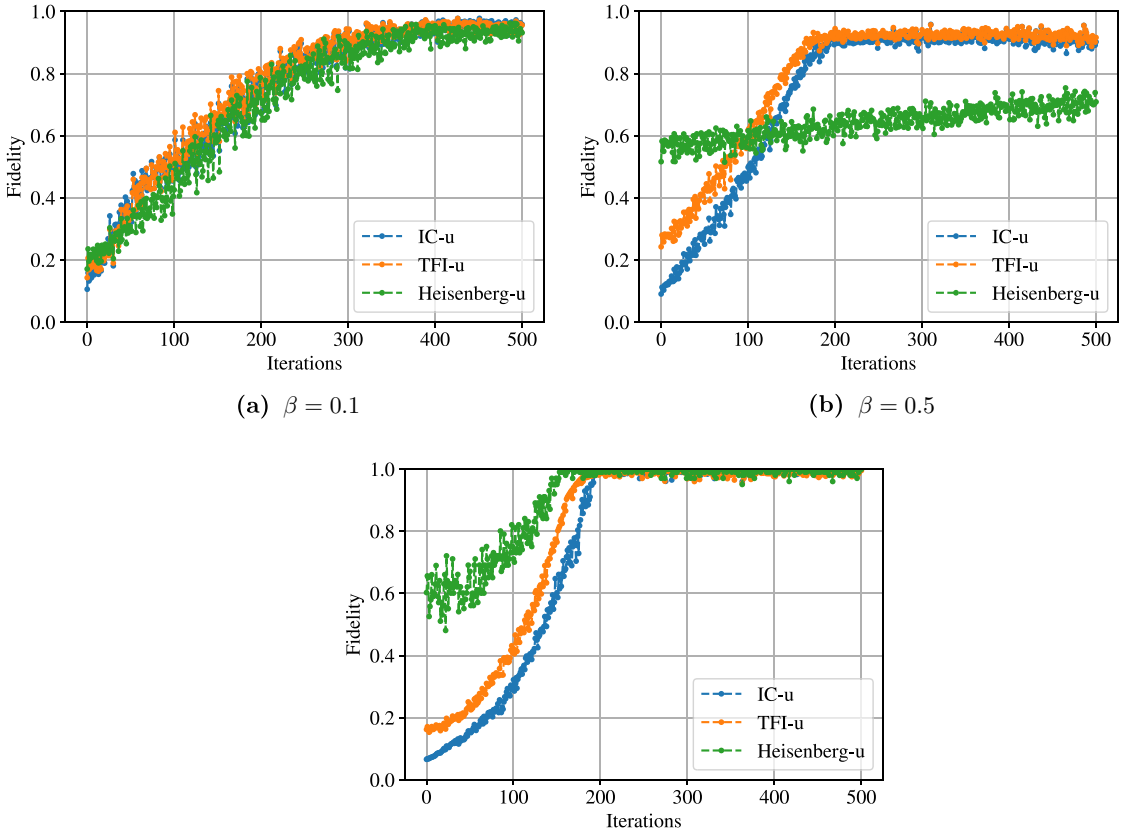
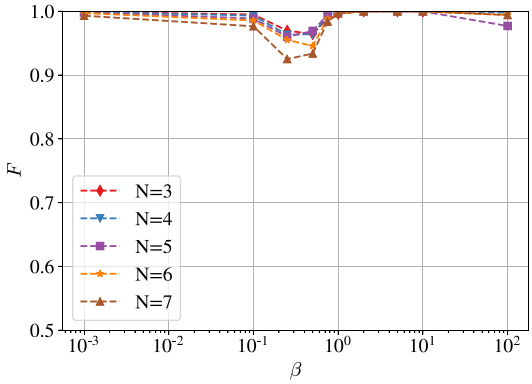


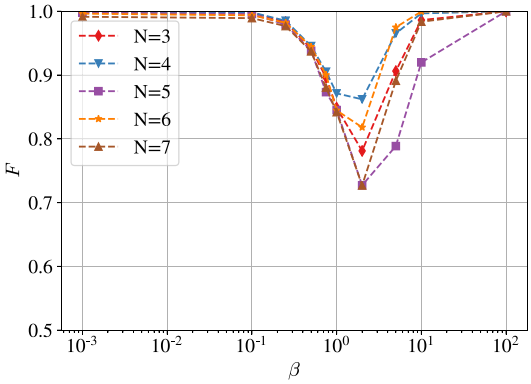
Figure 3. Optimization curves for the three models with uniform coefficients and $N = 4$. We observe in all the cases a constant increase of the fidelity, showing that minimizing the approximate free energy cost function tends to result in a maximization of the fidelity. It also shows that the final result found by the algorithm is always significantly better than the random initialization.

1. The optimization curves presented in Fig. 3 show that the optimization procedure improves the solution compared to a random initialization, both when a very high fidelity is reached at the end and when the fidelity is lower. It eliminates the possibility that random states being closed to the desired thermal states would explain our results. Moreover, the fidelity tends to increase with the number of iterations, showing that our approximate cost-function might be well-suited to our optimization goal.
2. Thermal states at low and high temperatures are easily approximated by our method, for all models and system sizes. Looking at the λ curves, we see that the optimizer is indeed able to find $\lambda = 0$ for very large β and $\lambda = 1$ for very low β. Hence, when the thermal state gets close to a maximally-mixed state or to a pure state, the algorithm learns to respectively maximize or minimize the noise, independently of the initial noise level.
3. The performance tends to degrade at intermediate temperatures, reaching for instance a fidelity of 0.6 for the Heisenberg model with random coefficients. However, there are several temperatures for which a non-trivial noise level is learned and the fidelity remains high, such as the same model at $\beta = 10^{-1}$, for which a fidelity above 96% is reached for all system sizes with a noise level between 0.5 and 0.8. Hence the algorithm can actually find the correct thermal state in non-trivial temperature regimes.

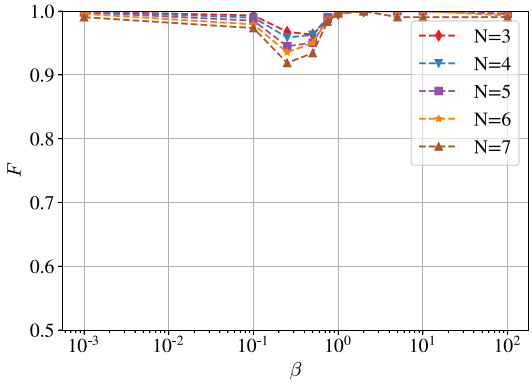
From those results, an important question to consider is whether the low fidelity obtained for some systems is due to a failure of the optimization procedure or to the potentially low expressibility of our noisy ansatz. To tackle this question, we tested different methods to optimize the parameters of the ansatz, including a grid-search in the parameter space for systems that are small enough to allow it to run in a reasonable time. We found no significant improvement in the fidelity compared to the original optimization method. We also tried to initialize the unitary ansatz to the ground-state solution before turning on the noise, but it did not result in a significant increase of fidelity neither. Finally, to evaluate the effect of our free energy approximation, we performed all the experiments previously mentioned using finite-difference on the true free energy. The corresponding results can be found in Supplementary Figure S2, where we observe very similar fidelities as with the approximate free energy method. It means that for the hardest systems tested in this work, the noisy ansatz was probably not expressible enough to output an accurate approximation of the thermal state, independently of the optimization algorithm.



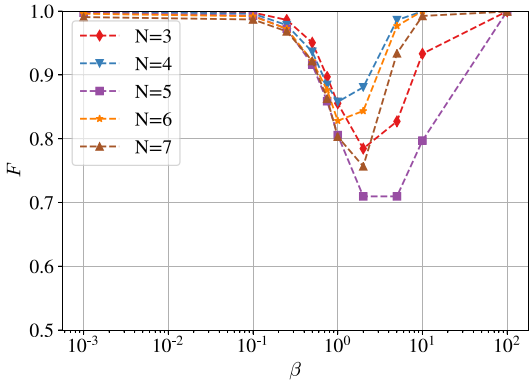
(a) Classical Ising chain with uniform coefficients.



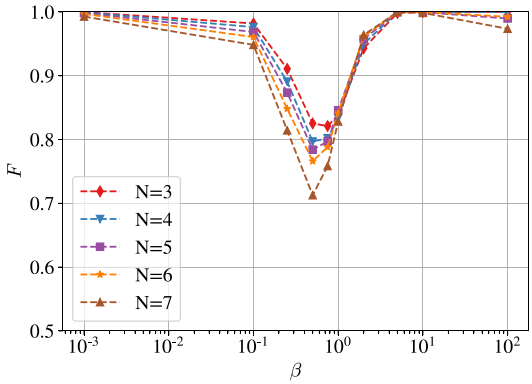
(b) Classical Ising chain with random coefficients.



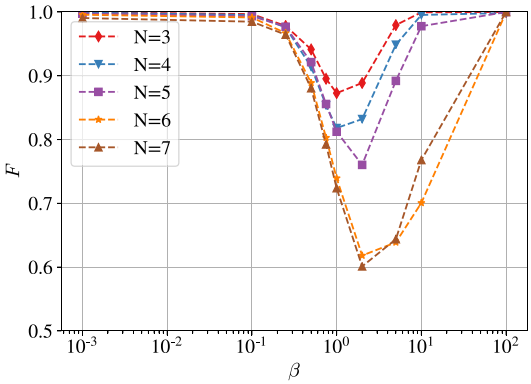
(c) Transverse-field Ising chain with uniform coefficients.



(d) Transverse-field Ising chain with random coefficients.



(e) 1D Heisenberg model with uniform coefficients.



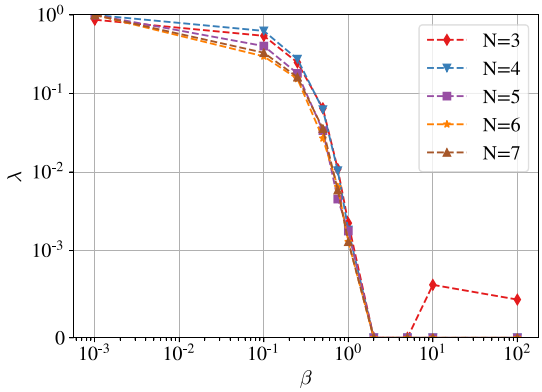
(f) 1D Heisenberg model with random coefficients.

Figure 4. Fidelities obtained using NAVQT as a function of the inverse temperature β, for various models and system sizes. For all the models, we observe that the algorithms reaches a high fidelity for low and high temperature, while it tends to decrease at intermediate temperatures. Overall, good performance is obtained at all temperatures for the two types of uniform Ising chains, while lower fidelities are reached with the other models.

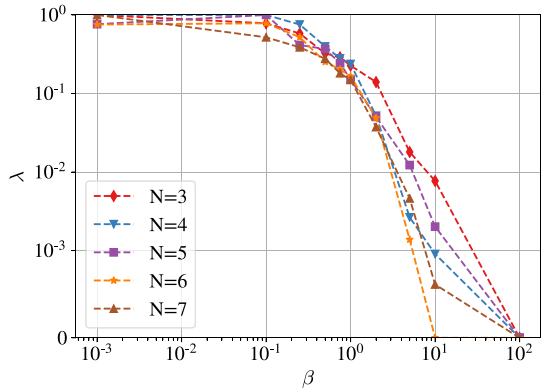
Changing the depolarizing gates to more general noise channels could help improve the expressibility of the ansatz and is let for future work.

Discussion

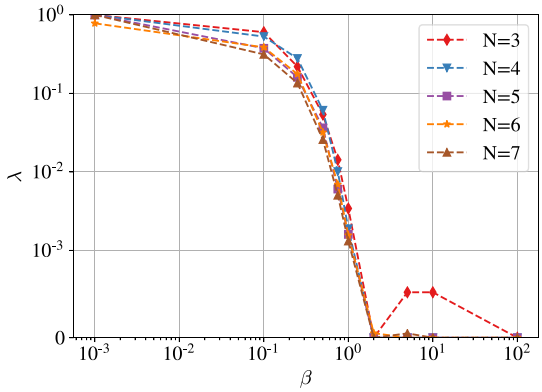
In this paper, we introduced a novel type of variational algorithms, in which the noise is parameterized and optimized together with the unitary gates. We used this architecture to prepare thermal states, overcoming some of the most common challenges for this task, such as the need of ancilla qubits and the adverse effect of noise. To optimize our ansatz, we used a closed-form approximation of the free-energy and performed gradient-descent with it. We investigated various Hamiltonians and deduced that the ability of our method to learn the correct thermal states strongly depends on the model, the temperature and the system size. While we systematically



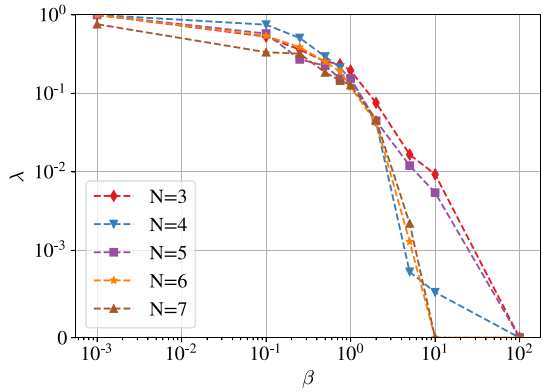
(a) Classical Ising chain with uniform coefficients.



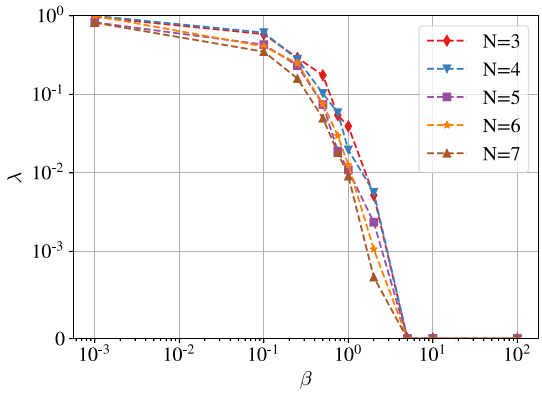
(b) Classical Ising chain with random coefficients.



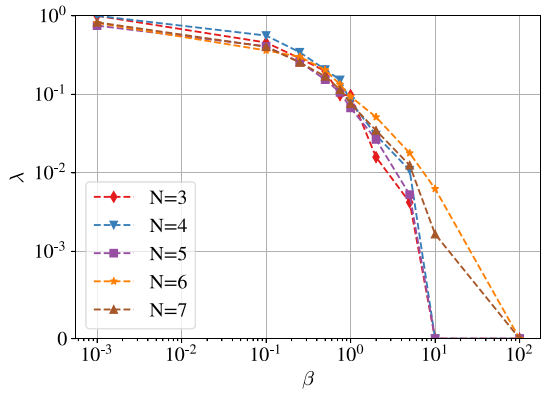
(c) Transverse-field Ising chain with uniform coefficients.



(d) Transverse-field Ising chain with random coefficients.



(e) 1D Heisenberg model with uniform coefficients.



(f) 1D Heisenberg model with random coefficients.

Figure 5. Final noise level λ as a function of the inverse temperature β for various models and system sizes. We used a *symlog* scale for the y-axis, hence the scale becomes linear below 10^{-3}. We observe a clear decrease of the noise level with β, with $\lambda \approx 1$ for $\beta = 10^{-3}$ (corresponding to the maximally-mixed state) and $\lambda \approx 0$ for $\beta \approx 10^2$ (corresponding to the ground-state). It shows that the general relationship between the noise and the temperature has overall been correctly learned by our model.

obtained fidelities above 0.9 for both the transverse-field and the classical Ising chain, we had fidelities below 0.7 at some temperatures for the 1D Heisenberg model with random coefficients. We also identified a specific range of temperatures for each model, for which the task is harder for NAVQT to solve. Our experiments with different optimization algorithms reveal that the failure of the ansatz to learn the correct thermal state in those cases is probably an expressibility rather than an optimization issue.

This paper serves as a starting point in the study of noise-assisted thermalization, and many avenues are still open for future work. For instance, we only considered a single shared parameter λ for all the depolarizing gates, as it allowed us to derive an approximation of the free energy, which simplified the optimization process. Varying the noise across each layer and each qubit independently could significantly increase the expressibility of

the ansatz. More generally, replacing the depolarizing gates by channels that are more tailored for thermal state preparation would be an interesting avenue to improve our method. For instance, Davies maps are non-unital channels that can model the evolution of quantum systems weakly-coupled to a thermal reservoir, making them particularly adapted to thermal state preparation[46]. Moreover, their unitary and dissipative parts commute, making the calculation of the entropy potentially easier than for our ansatz.

A second important aspect for future work would be to better understand the theory behind noise-assisted variational circuits. For instance, what are the conditions on the Hamiltonian and the temperature under which NAVQT can approximate the thermal state with an arbitrary high fidelity? How does our method scale with the system size? What type of noise is necessary to approximate a given thermal state?

Finally, it could be interesting to study the optimization landscape of NAVQT and potentially come up with optimization algorithms that are more tailored to this problem. For instance, it has been shown that a barren plateau phenomenon occurs in noisy circuits that are similar to our ansatz[47]. It can potentially hinder the scalability of our method, as it relies explicitly on increasing the noise. Finding the relationship between the temperature β, the system size N and the magnitude of the gradient could be an interesting direction for future research.

Data availability

All code is available at https://github.com/jfold/navqt.

Received: 8 November 2021; Accepted: 10 February 2022
Published online: 09 March 2022

References

1. Verstraete, F., Wolf, M. M. & Cirac, J. I. Quantum computation and quantum-state engineering driven by dissipation. *Nat. Phys.* **5**, 633–636 (2009).
2. Kübler, J. M., Arrasmith, A., Cincio, L. & Coles, P. J. An adaptive optimizer for measurement-frugal variational algorithms. *Quantum* **4**, 263 (2020).
3. Sweke, R. *et al.* Stochastic gradient descent for hybrid quantum-classical optimization. *Quantum* **4**, 314 (2020).
4. Cao, C. & Wang, X. Noise-assisted quantum autoencoder. arXiv preprint arXiv arXiv:2012.08331 (2020).
5. Spohn, H. & Lebowitz, J. L. Irreversible thermodynamics for quantum systems weakly coupled to thermal reservoirs. *Adv. Chem. Phys.* **38**, 109–142 (1978).
6. Cao, Y. *et al.* Quantum chemistry in the age of quantum computing. *Chem. Rev.* **119**, 10856–10915 (2019).
7. Whitfield, J. D. *et al.* Introduction to quantum algorithms for physics and chemistry. *Advances in Chemical Physics* (2013).
8. Lee, C.-K., Patil, P., Zhang, S. & Hsieh, C.-Y. A neural-network variational quantum algorithm for many-body dynamics. arXiv preprint arXiv:2008.13329 (2020).
9. Brandao, F. G. & Svore, K. M. Quantum speed-ups for solving semidefinite programs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, 415–426 (IEEE, 2017).
10. Van Apeldoorn, J., Gilyén, A., Gribling, S. & de Wolf, R. Quantum sdp-solvers: Better upper and lower bounds. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, 403–414 (IEEE, 2017).
11. Kieferová, M. & Wiebe, N. Tomography and generative training with quantum Boltzmann machines. *Phys. Rev. A* **96**, 062327 (2017).
12. Amin, M. H., Andriyash, E., Rolfe, J., Kulchytskyy, B. & Melko, R. Quantum Boltzmann machine. *Phys. Rev. X* **8**, 021050 (2018).
13. Jerrum, M. & Sinclair, A. Polynomial-time approximation algorithms for the Ising model. *SIAM J. Comput.* **22**, 1087–1116 (1993).
14. Eldan, R., Koehler, F. & Zeitouni, O. A spectral condition for spectral gap: Fast mixing in high-temperature Ising models. arXiv preprint arXiv:2007.08200 (2020).
15. Wu, J. & Hsieh, T. H. Variational thermal quantum simulation via thermofield double states. *Phys. Rev. Lett.* **123**, 220502 (2019).
16. Zhu, D. *et al.* Generation of thermofield double states and critical ground states with a quantum computer. arXiv preprint arXiv:1906.02699 (2019).
17. Zoufal, C., Lucchi, A. & Woerner, S. Variational quantum Boltzmann machines. arXiv preprint arXiv:2006.06004 (2020).
18. Wang, Y., Li, G. & Wang, X. Variational quantum Gibbs state preparation with a truncated taylor series. arXiv preprint arXiv:2005.08797 (2020).
19. Verdon, G., Broughton, M. & Biamonte, J. A quantum algorithm to train neural networks using low-depth circuits. arXiv preprint arXiv:1712.05304 (2017).
20. Martyn, J. & Swingle, B. Product spectrum ansatz and the simplicity of thermal states. *Phys. Rev. A* **100**, 032107 (2019).
21. Verdon, G., Marks, J., Nanda, S., Leichenauer, S. & Hidary, J. Quantum Hamiltonian-based models and the variational quantum thermalizer algorithm. arXiv preprint arXiv:1910.02071 (2019).
22. Chowdhury, A. N., Low, G. H. & Wiebe, N. A variational quantum algorithm for preparing quantum Gibbs states. arXiv preprint arXiv:2002.00055 (2020).
23. Franca, D. S. & Garcia-Patron, R. Limitations of optimization algorithms on noisy quantum devices. arXiv preprint arXiv:2009.05532 (2020).
24. Temme, K., Bravyi, S. & Gambetta, J. M. Error mitigation for short-depth quantum circuits. *Phys. Rev. Lett.* **119**, 180509 (2017).
25. Endo, S., Cai, Z., Benjamin, S. C. & Yuan, X. Hybrid quantum-classical algorithms and quantum error mitigation. arXiv preprint arXiv:2011.01382 (2020).
26. Cottrell, W., Freivogel, B., Hofman, D. M. & Lokhande, S. F. How to build the thermofield double state. *J. High Energy Phys.* **2019**, 58 (2019).
27. D'Hoker, E., Dong, X. & Wu, C.-H. An alternative method for extracting the von Neumann entropy from Rényi entropies. *J. High Energy Phys.* **2021**, 1–23 (2021).
28. McArdle, S. *et al.* Variational ansatz-based quantum simulation of imaginary time evolution. *NPJ Quantum Inf.* **5**, 1–6 (2019).
29. Benedetti, M., Fiorentini, M. & Lubasch, M. Hardware-efficient variational quantum algorithms for time evolution. *Phys. Rev. Res.* **3**, 033083 (2021).
30. Shingu, Y. *et al.* Boltzmann machine learning with a variational quantum algorithm. arXiv preprint arXiv:2007.00876 (2020).
31. Sun, S.-N. *et al.* Quantum computation of finite-temperature static and dynamical properties of spin systems using quantum imaginary time evolution. *PRX Quantum* **2**, 010317 (2021).
32. Motta, M. *et al.* Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution. *Nat. Phys.* **16**, 205–210 (2020).
33. Schuld, M., Bergholm, V., Gogolin, C., Izaac, J. & Killoran, N. Evaluating analytic gradients on quantum hardware. *Phys. Rev. A* **99**, 032331 (2019).

34. Team, Q. A. & collaborators. Cirq. https://doi.org/10.5281/zenodo.4062499 (2020).
35. Broughton, M. *et al.* Tensorflow quantum: A software framework for quantum machine learning. arXiv preprint arXiv:2003.02989 (2020).
36. Farhi, E., Goldstone, J. & Gutmann, S. A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028 (2014).
37. Wierichs, D., Gogolin, C. & Kastoryano, M. Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer. *Phys. Rev. Res.* **2**, 043246 (2020).
38. Morales, M. E., Biamonte, J. D. & Zimborás, Z. On the universality of the quantum approximate optimization algorithm. *Quantum Inf. Process.* **19**, 1–26 (2020).
39. Preskill, J. Lecture notes for physics 229: Quantum information and computation. *Calif. Inst. Technol.* **16**, 1–8 (1998).
40. Blanchard, P., Higham, D. J. & Higham, N. J. Accurately computing the log-sum-exp and softmax functions. *IMA J. Numer. Anal.* **41**, 2311–2330 (2021).
41. Billoni, O. V., Cannas, S. A. & Tamarit, F. A. Spin-glass behavior in the random-anisotropy Heisenberg model. *Phys. Rev. B* **72**, 104407 (2005).
42. Gong, S.-S., Zhu, W. & Sheng, D. Emergent chiral spin liquid: Fractional quantum hall effect in a Kagome Heisenberg model. *Sci. Rep.* **4**, 1–6 (2014).
43. Jepsen, P. N. *et al.* Spin transport in a tunable Heisenberg model realized with ultracold atoms. *Nature* **588**, 403–407 (2020).
44. Rodriguez-Nieva, J. F. Turbulent relaxation after a quench in the Heisenberg model. *Phys. Rev. B* **104**, L060302 (2021).
45. Powers, C., Bassman, L. & de Jong, W. A. Exploring finite temperature properties of materials with quantum computers. arXiv preprint arXiv:2109.01619 (2021).
46. Roga, W., Fannes, M. & Życzkowski, K. Davies maps for qubits and qutrits. *Rep. Math. Phys.* **66**, 311–329 (2010).
47. Wang, S. *et al.* Noise-induced barren plateaus in variational quantum algorithms. arXiv preprint arXiv:2007.14384 (2020).

Acknowledgements
We would like to thank Michael Kastoryano, Jonatan Bohr Brask and Daniel Stilck França for fruitful discussions during this work, as well as Dan Browne for useful discussions and feedback during the writing of this manuscript. We would also like to thank Guillaume Verdon and Antonio Martinez for providing helpful tutorials and advice on noisy simulations of variational quantum circuits with Tensorflow Quantum. AP was supported by the Engineering and Physical Sciences Research Council [Grant Number EP/S021582/1]. JF was supported by the William Demant Foundation [Grant Number 18-4438].

Author contributions
J.F. and A.P. formulated, conceived and conducted the theory and experiments. J.F., A.P. and L.K.H. analyzed the results. All authors reviewed the manuscript.

Competing interests
The authors declare no competing interests.

Additional information
Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-022-07296-z.

Correspondence and requests for materials should be addressed to J.F.

Noise-Assisted Variational Quantum Thermalization

Jonathan Foldager[1*], Arthur Pesah[2], and Lars Kai Hansen[1]

[1]Technical University of Denmark, Department for Applied Mathematics and Computer Science, Kongens Lyngby, 2800, Denmark
[2]University College London, Department of Physics and Astronomy, London WC1E 6BT, United Kingdom
[*]jonf@dtu.dk

SUPPLEMENTARY INFORMATION

Supplementary Note: Estimating the free energy of a noisy circuit

In order to learn the thermal state with NAVQT, we need to minimize the free energy. Obtaining the free energy from the output of a quantum circuit is hard, since the entropy is a highly non-linear function of the state. For unitary evolutions, the entropy is constant, but for non-unitary circuits, including our ansatz, the entropy needs to be estimated for each change of parameters. To simplify this task, we consider the following approximation, represented in Fig. 1c of the main manuscript: all the depolarizing gates are shifted to the beginning of the circuit. Using this circuit, it is now possible to compute the entropy analytically.

If m is the number of layers of our unitary ansatz, the approximate circuit consists in the composition of m depolarizing gates $\mathscr{D}(\lambda)$ for each qubit. We can now use the fact that the composition of depolarizing gates is itself a depolarizing gate:

$$\mathscr{D}(\lambda_2)\circ\mathscr{D}(\lambda_1) = \mathscr{D}(1-(1-\lambda_2)(1-\lambda_1)) \tag{S1}$$

or more generally

$$\mathscr{D}(\lambda_m)\circ\ldots\circ\mathscr{D}(\lambda_1) = \mathscr{D}(1-(1-\lambda_m)\ldots(1-\lambda_1)) \tag{S2}$$

Assuming that the noise parameter λ is the same for all the gates, then the above simplifies to $\mathscr{D}(1-(1-\lambda)^m)$. If we note $\Lambda = 1-(1-\lambda)^m$ this new parameter, the entropy of m consecutive noise gates acting on a single qubit initialized with $|0\rangle$ can be written as

$$\begin{aligned} S(\rho_\Lambda) &= -\operatorname{Tr}[\rho_\Lambda \ln(\rho_\Lambda)] \\ &= -\operatorname{Tr}\left[\left((1-\Lambda)\,|0\rangle\langle 0| + \Lambda\frac{\mathbb{1}}{d}\right)\ln\left((1-\Lambda)\,|0\rangle\langle 0| + \Lambda\frac{\mathbb{1}}{d}\right)\right] \\ &= -\left[\left((1-\Lambda)+\frac{\Lambda}{d}\right)\ln\left((1-\Lambda)+\frac{\Lambda}{d}\right) + \frac{(d-1)\Lambda}{d}\ln\left(\frac{\Lambda}{d}\right)\right] \end{aligned} \tag{S3}$$

and substituting for $\Lambda = 1-(1-\lambda)^m$

$$\begin{aligned} S(\rho_\lambda) = -\Bigg[&\left((1-\lambda)^m + \frac{(1-(1-\lambda)^m)}{d}\right)\ln\left((1-\lambda)^m + \frac{(1-(1-\lambda)^m)}{d}\right) \\ &+ \frac{(d-1)(1-(1-\lambda)^m)}{d}\ln\left(\frac{(1-(1-\lambda)^m)}{d}\right)\Bigg] \end{aligned} \tag{S4}$$

We now use the fact the entropy of a product state is the sum of the individual entropies to write

$$S(\rho_\lambda^{\otimes N}) = NS(\rho_\lambda) \tag{S5}$$

which directly gives us the entropy of the state preceding the unitary ansatz. Since applying a unitary operation to a state does not change its entropy, it means that the overall entropy of the output state, that we call $S(\lambda)$, is given by the expression above. To optimize over it, we need to compute its gradient, which can also be obtained analytically as

$$\nabla_\lambda S(\lambda) = N\frac{d-1}{d}m(1-\lambda)^{m-1}\left[-\ln\left(\frac{(1-(1-\lambda)^m)}{d}\right) + \ln\left(\frac{(1-(1-\lambda)^m + d(1-\lambda)^m)}{d}\right)\right] \tag{S6}$$

The overall free energy, which we want to minimize, is given by

$$F(\boldsymbol{\theta},\lambda) = E(\boldsymbol{\theta},\lambda) - TS(\lambda). \tag{S7}$$

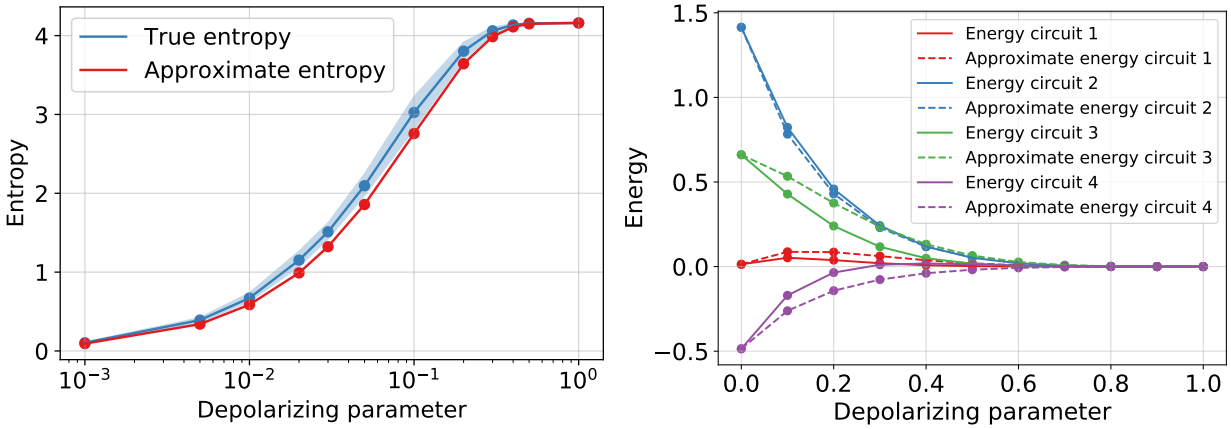
The gradient of the energy with respect to θ can be efficiently computed on a quantum device using the parameter shift-rule, while its gradient with respect to λ can be computed using finite-difference (since $E(\lambda)$ itself can easily be extracted from the output of the circuit). Therefore, the overall gradient, given by

$$\nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta},\lambda) = \nabla_{\boldsymbol{\theta}} E(\boldsymbol{\theta},\lambda) \tag{S8}$$

$$\nabla_{\lambda} F(\boldsymbol{\theta},\lambda) = \nabla_{\lambda} E(\boldsymbol{\theta},\lambda) - T\nabla_{\lambda} S(\lambda) \tag{S9}$$

can be efficiently computed using our circuit approximation.

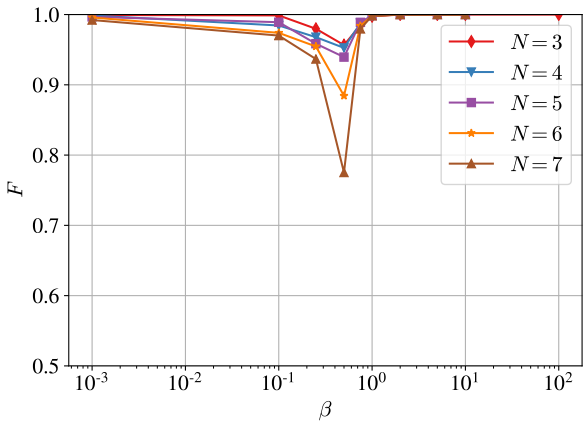
Supplementary Figures



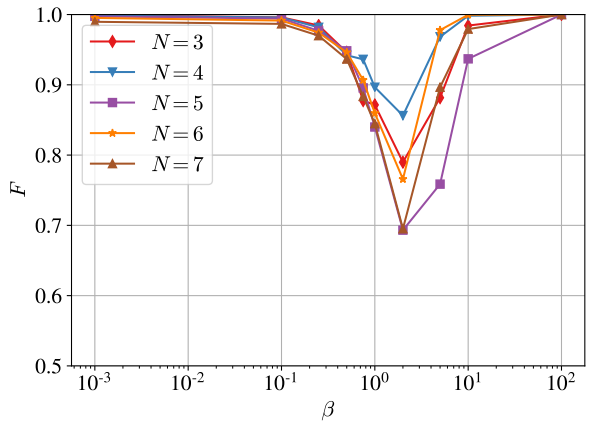
(a) Entropy comparison

(b) Energy comparison

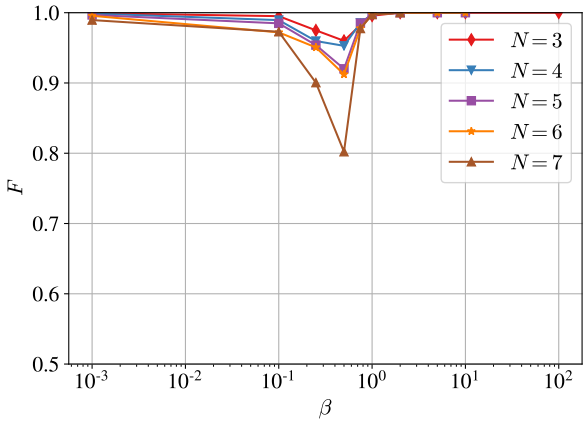
Figure S1. Comparison of the approximate ansatz with the true one, for both the entropy and the energy as a function of the depolarizing noise, for random circuits with 6 qubits and 3 layers. **(a)** Entropy of the two circuit types. Since the entropy of the true circuit depends on the unitary parameters, we sampled 100 random parameters and took the average, minimum and maximum of the entropy (blue area). We see that the two curves follow a similar trajectory, with the approximate entropy being a lower bound on the true one. **(b)** Energy of the two circuit types for the transverse-field Ising model with uniform coefficients. Each color represents a circuit with different random unitary parameters. We see that the approximate energy tends to be close to the true one, following an overall similar trajectory.



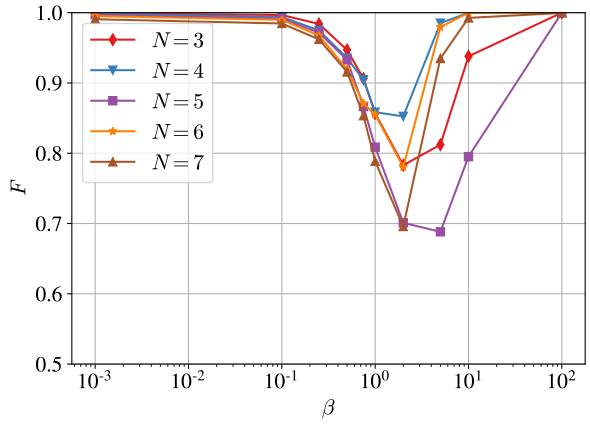
(a) IC with uniform coefficients.



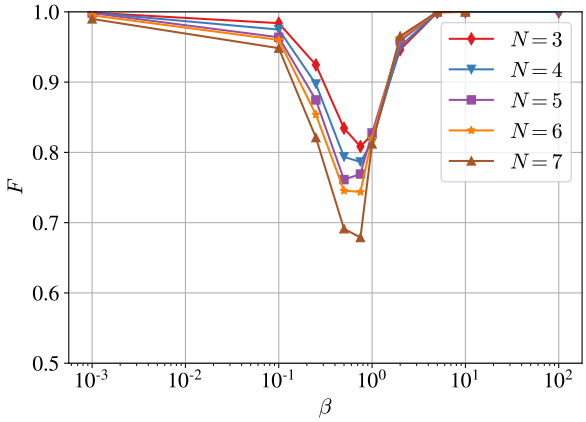
(b) IC with random coefficients.



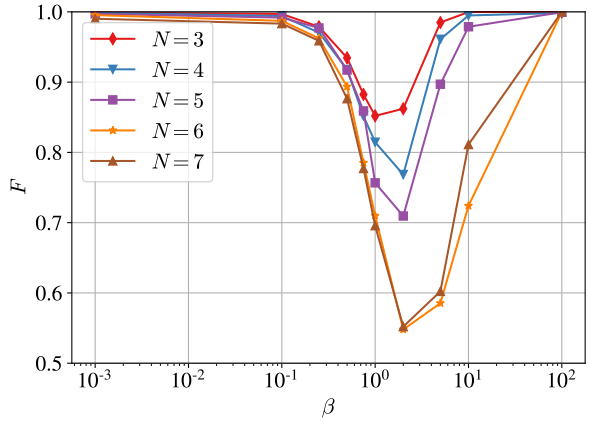
(c) TFI with uniform coefficients.



(d) TFI with random coefficients.



(e) Heisenberg with uniform coefficients.



(f) Heisenberg with random coefficients.

Figure S2. Fidelities obtained when minimizing the actual free energy (as opposed to the approximation), using finite-difference to obtain the gradient. We see that the results are similar to those in Fig. 3 of the main manuscript, where the approximate free energy is used for optimization. It shows that the approximation serves as a good heuristics for optimizing our NAVQT ansatz.

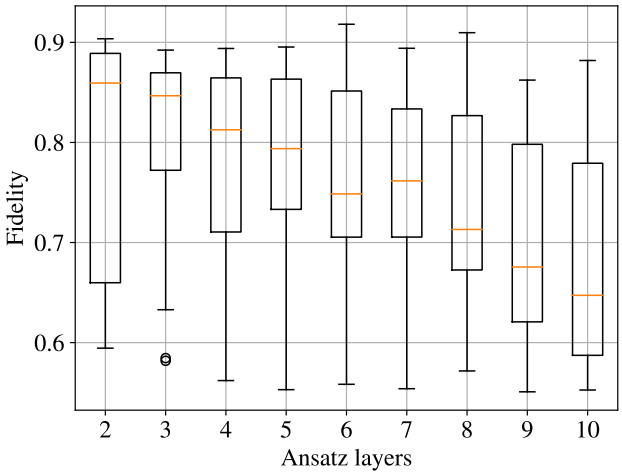


Figure S3. Thermal state fidelity as a function of ansatz layers for 3 qubits Heisenberg model with random coefficients over 5 random seeds. It illustrates that fidelity does not increase with ansatz layers, which suggest that it is not lack of circuit expressivity, but instead either lack of precision in the the free energy approximation and/or that the depolarization channel is not enough for thermalization. A similar pattern to this figure was kept for all the systems we studied. We were able to improve the fidelity slightly when comparing to Fig. 4f ($N = 3$), but we also see that the median fidelity decreases as a function of layers, which we hypothesize this is due to the entropy approximation gets worse with the number of layers and/or that the depolarization noise accumulates.

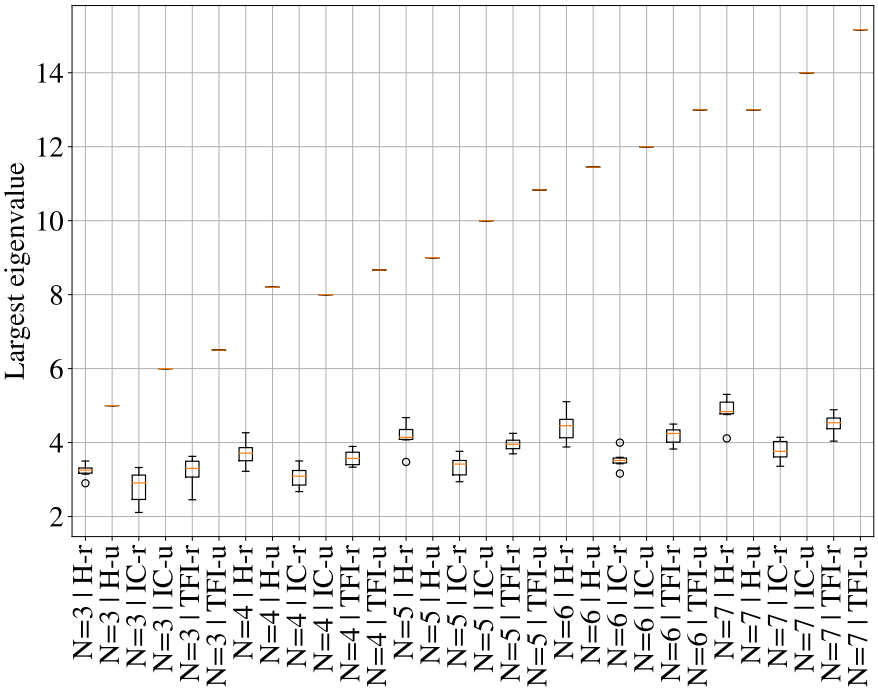


Figure S4. Largest eigenvalue for the models studied in the paper over five seeds. Here, H, IC and TFI refers to the Heisenberg, Ising Chain and Transvers-field Ising chain models, respectively. The "-u" and "-r" suffix refers to if the coefficients are uniform (equal to one) or randomly drawn from a standard normal distribution.

Appendix B

Paper B (under review)

THE FOLLOWING PAGES CONTAIN A COPY OF THE PREPRINT VERSION OF THE PAPER CURRENTLY UNDER REVIEW

Can shallow quantum circuits scramble local noise into global white noise?

Jonathan Foldager[1,2,*] and Bálint Koczor[2,3,†]

[1] Department of Applied Mathematics and Computer Science, Technical University of Denmark, Denmark
[2] Department of Materials, University of Oxford, Parks Road, Oxford OX1 3PH, United Kingdom
[3] Quantum Motion, 9 Sterling Way, London N7 9HJ, United Kingdom

Shallow quantum circuits are believed to be the most promising candidates for achieving early practical quantum advantage – this has motivated the development of a broad range of error mitigation techniques whose performance generally improves when the quantum state is well approximated by a global depolarising (white) noise model. While it has been crucial for demonstrating quantum supremacy that random circuits scramble local noise into global white noise—a property that has been proved rigorously—we investigate to what degree practical shallow quantum circuits scramble local noise into global white noise. We define two key metrics as (a) density matrix eigenvalue uniformity and (b) commutator norm. While the former determines the distance from white noise, the latter determines the performance of purification based error mitigation. We derive analytical approximate bounds on their scaling and find in most cases they nicely match numerical results. On the other hand, we simulate a broad class of practical quantum circuits and find that white noise is in certain cases a bad approximation posing significant limitations on the performance of some of the simpler error mitigation schemes. On a positive note, we find in all cases that the commutator norm is sufficiently small guaranteeing a very good performance of purification-based error mitigation. Lastly, we identify techniques that may decrease both metrics, such as increasing the dimensionality of the dynamical Lie algebra by gate insertions or randomised compiling.

I. INTRODUCTION

Current generations of quantum hardware can already significantly outperform classical computers in random sampling tasks [1, 2] and hopefully future hardware developments will enable powerful applications in quantum machine learning [3], fundamental physics [4, 5] and in developing novel drugs and materials [6–9]. The scale and precision of the technology today is, however, still below what is required for fully fault-tolerant quantum computation: Due to noise accumulation in the noisy intermediate-scale quantum (NISQ) era [10], one is thus limited to only shallow-depth quantum circuits which led to the development of a broad range of hybrid quantum-classical protocols and quantum machine learning algorithms [11–13].

The aim in this paradigm is to prevent excessive error buildup via a parameterised, shallow-depth quantum circuit and then perform a series of repeated measurements in order to extract expected values. These expected values are then post processed on a classical computer in order to update the parameters of the circuits, e.g., as part of a training procedure. A major challenge is the potential need for an excessive number of circuit repetitions which, however, can be significantly suppressed by the use of advanced training algorithms [14–16] or via classical-shadows-based protocols [17–19]. As such, the primary limitation of near-term applications is the damaging effect of gate noise on the estimated expected values which can only be reduced by advanced error mitigation techniques [12, 20].

Error mitigation comprises a broad collection of diverse techniques that generally aim to estimate precise expected values by suppressing the effect of hardware imperfections [12, 20]. Due to the diversity of techniques and due to the significant differences in the range of applicability, the need for performance metrics was recently emphasised [20]. This motivates the present work to characterise noise in typical practical circuits, e.g., in quantum simulation or in quantum machine learning, and define two key metrics that determine the performance of a broad class of error mitigation techniques: (a) eigenvalue uniformity as a closeness to global depolarising (white) noise and (b) norm of the commutator between the ideal and noisy quantum states. While (b) determines the performance of purification based error mitigation techniques [21, 22], (a) implies a good performance of all error mitigation techniques.

Our primary motivation is that gate errors in complex quantum circuits are scrambled into global white noise [1, 23]. This property has been proved for random circuits by establishing exponentially decreasing error bounds; surprisingly, in our numerical simulations we find that in many practical scenarios the same bounds apply relatively well. In particular, we find that both our metrics, (a) the distance from global-depolarising noise and (b) the commutator norm, are approximated as

$$f(\nu) = \alpha \frac{e^{-\xi}\xi}{(1-e^{-\xi})\sqrt{\nu}} = \frac{\alpha}{\sqrt{\nu}} + O(\xi), \qquad (1)$$

where ν is the number of gates in the quantum circuit, ξ is the number of expected errors in the entire circuit and α is a constant. As such, if one keeps the error rate small $\xi \ll 1$ but increases the number of gates in a circuit then both (a) and (b) are expected to decrease. This is a highly desirable property in practice, e.g., white noise

* jonf@dtu.dk
† balint.koczor@materials.ox.ac.uk

does not introduce a bias to the expected-value measurement but only a trivial, global scaling as we detail in the rest of this introduction.

In the present work we simulate a broad range of quantum circuits often used in practice and identify scenarios where this approximation holds well, by means of gate parameters and circuit structures are sufficiently random. We also identify strategies that improve scrambling local gate noise into global white noise, such as inserting additional gates into a circuit to increase the dimensionality of its Lie algebra [24]. In most cases, however, we conclude that white noise is not necessarily a good approximation due to the large prefactor α in Eq. (1). Thus the performance of some error mitigation techniques that rely on a global-depolarising noise assumption is limited. On the other hand, we find that in all cases the commutator norm, our other key metric, is smaller by at least 1-2 orders of magnitude guaranteeing a very good performance of purification-based error mitigation techniques.

Our work is structured as follows. In the rest of this introduction we briefly review global depolarising noise and how it can be exploited in error mitigation, and then briefly review purification-based error mitigation techniques and their performance. In Section II we introduce theoretical notions and finally in Section III we present our simulation results.

A. Global depolarisation and error mitigation

In the NISQ-era, we don't have comprehensive solutions to error correction, which has led the field to develop error mitigation techniques. These techniques aim to extract expected values $\langle O\rangle_{ideal} := \mathrm{tr}[O\rho_{id}]$ of observables—typically some Hamiltonian of interest—with respect to an ideal noiseless quantum state ρ_{id}.

A very simple error model, the global depolarising noise channel, has been very often considered as a relatively good approximation to complex quantum circuits. For qubit states, the channel mixes the ideal, noise-free state with the maximally mixed state Id/d of dimension $d = 2^N$ as

$$\rho_{wn} := \eta\rho_{id} + (1-\eta)\mathrm{Id}/d. \tag{2}$$

Here $\eta \approx F$ is a probability that approximates the fidelity as $F = \eta + (1-\eta)/d$. The white noise channel has been commonly used in the literature for modelling errors in near-term quantum computers [25] and, in particular, it has been shown to be a very good approximation to noise in random circuits [1, 23]. White noise is extremely convenient as it lets the user extract, after rescaling by η, the ideal expected value of any traceless Hermitian observable O via

$$\langle O\rangle_{ideal} = \mathrm{tr}[O\rho_{wn}]/\eta. \tag{3}$$

Of course, for small fidelities $\eta \ll 1$ the expected value $\mathrm{tr}[O\rho_{wn}]$ requires a significantly increased sampling to suppress shot noise. In this model, the scaling factor η is a global property and can be estimated experimentally, e.g., via randomised measurements [25], via extrapolation [26] or via learning-based techniques [27].

Global depolarisation, however, may not be sufficiently accurate model to capture more subtle effects of gate noise and thus rescaling an experimentally estimated expected value yields a biased estimate of the ideal one as $\langle O\rangle_{bias} := \mathrm{tr}[O\rho]/\eta - \langle O\rangle_{ideal}$. The bias here $\langle O\rangle_{bias}$ is not a global property, i.e., it is specific to each observable, and requires the use of more advanced error mitigation techniques to suppress.

Intuitively, one expects the bias is small for quantum states that are well approximated by a global depolarising model as $\rho \approx \rho_{wn}$ and, indeed, we find a general upper bound in terms of the trace distance as

$$|\langle O\rangle_{bias}| = \frac{|\,\mathrm{tr}[O\rho] - \mathrm{tr}[O\rho_{wn}]|}{\eta} \leq \frac{\|O\|_\infty \|\rho - \rho_{wn}\|_1}{\eta}. \tag{4}$$

Here $\|O\|_\infty$ is the operator norm as the absolute largest eigenvalue of the traceless O, refer to ref. [28] for a proof. As such, a small trace distance guarantees a small bias and thus indirectly determines the performance of all error mitigation techniques – and further protocols [19, 29].

In this work, we characterise how close noisy quantum states ρ in practical applications approach white noise states ρ_{wn} and consider various types of variational quantum circuits that are typical for NISQ applications. When the above trace distance is small then it guarantees a small bias in expected values which allows us to nearly trivially mitigate the effect of gate noise, i.e., via a simple rescaling.

B. Purification-based error mitigation and the commutator norm

Another core metric we will consider is the commutator norm between the ideal and noisy quantum states as $\mathcal{E}_C := \|[\rho_{id}, \rho]\|_1$, which determines the performance of purification based error mitigation techniques [28] – a small commutator norm has significant practical implications as it guarantees that one can accurately determine expected values using the ESD/VD [21, 22] error mitigation techniques. In particular, independently preparing n copies of the noisy quantum state and applying a derangement circuit to entangle the copies, allows one to estimate the expected value

$$\frac{\mathrm{tr}[\rho^n O]}{\mathrm{tr}[\rho^n]} = \langle O\rangle_{ideal} + \mathcal{E}_{ESD}.$$

The approach is very NISQ-friendly [30, 31] and its approximation error $\mathcal{E}_{ESD}$ approaches in exponential order a noise floor as we increase the number of copies n [21]; This noise floor is determined generally by the commutator norm $\mathcal{E}_C$ but in the most typical applications of preparing eigenstates, the noise floor is quadratically smaller as $\mathcal{E}_C^2$ [28].

Note that this commutator can vanish even if the quantum state is very far from a white noise state, in fact it generally vanishes when ρ_{id} approximates an eigenvector of ρ. When a state is close to the white noise approximation then a small commutator norm is guaranteed, however, we demonstrate that the latter is a much less stringent condition and a much better approximation in practice than the former: in all instances we find that the commutator norm is significantly smaller than the trace distance from white noise states.

II. THEORETICAL BACKGROUND

In this section we introduce the main theoretical notations and recapitulate the most relevant results from the literature.

A. General properties of noisy quantum states

Recall that any quantum state of dimension d can be represented via its density matrix ρ that generally admits the spectral decomposition as

$$\rho = \sum_{k=1}^{d} \lambda_k \left|\psi_k\right\rangle\!\left\langle\psi_k\right|, \tag{5}$$

where we focus on N-qubit systems of dimension $d = 2^N$. Here λ_k are non-negative eigenvalues and $|\psi_k\rangle$ are eigenvectors. Since $\sum_i \lambda_i = 1$, the spectrum $\underline{\lambda}$ is also interpreted as a probability distribution.

If ρ is prepared by a perfect, noise-free unitary circuit, only one eigenvalue is different from zero and the corresponding eigenvector is the ideal quantum state as $|\psi_{id}\rangle$. In contrast, an imperfect circuit prepares a ρ that has more than one non-zero eigenvalues and is thus a probabilistic mixture of the pure quantum states $|\psi_k\rangle$, e.g., due to interactions with a surrounding environment. In fact, noisy quantum circuits that we typically encounter in practice produce quite particular structure of the eigenvalue distribution: one dominant component that approximates the ideal quantum state $|\psi_1\rangle \approx |\psi_{id}\rangle$ mixed with an exponentially growing number of "error" eigenvectors that have small eigenvalues. White noise is the limiting case where non-dominant eigenvalues are exponentially small $\propto 1/d$ and $|\psi_1\rangle \approx |\psi_{id}\rangle$.

The quality of the noisy quantum state is then defined by the probability of the ideal quantum state as the fidelity $F := \langle\psi_{id}|\rho|\psi_{id}\rangle$; We show in Appendix A that for any quantum state it approaches the dominant eigenvalue λ_1 as

$$\lambda_1 = F + O(\mathcal{E}_C), \tag{6}$$

where we compute the error term analytically in terms of the commutator norm $\mathcal{E}_C = \|[\rho_{id}, \rho]\|_1$ from Section I A. This property is actually completely general and applies to any density matrix.

B. Practically motivated noise models

Most typical noise models used in practice, such as local depolarising or dephasing noise, admit the following probabilistic interpretation: a noisy gate operation $\Phi(\rho)$ can be interpreted as a mixture of the noise-free operation U that happens with probability $1-\epsilon$ and an error contribution as

$$\Phi_k(\rho) = (1-\epsilon)U_k\rho U_k^\dagger + \epsilon\Phi_{err}(U_k\rho U_k^\dagger). \tag{7}$$

Here U_k is the kth ideal quantum gate and the completely positive trace-preserving (CPTP) map Φ_{err} happens with probability ϵ and accounts for all error events during the execution of a gate. A quantum circuit is then a composition of a series of ν such quantum gates which prepares the convex combination as

$$\rho = \eta\rho_{id} + (1-\eta)\rho_{err}. \tag{8}$$

Here $\rho_{id} := |\psi_{id}\rangle\langle\psi_{id}|$ is the ideal noise-free state, ρ_{err} is an error density matrix and $\eta = (1-\epsilon)^\nu$ is the probability that none of the gates have undergone errors. This probability actually [23, 28] approximates the fidelity $F := \langle\psi_{id}|\rho|\psi_{id}\rangle$ given the noise model in Eq. (7) as

$$F = (1-\epsilon)^\nu + \mathcal{E}_F = e^{-\xi} + \mathcal{E}_F + O(\epsilon^2/\nu). \tag{9}$$

Here we approximate $(1-\xi/\nu)^\nu = e^{-\xi} + O(\epsilon^2/\nu)$ for small ϵ and large ν where $\xi := \epsilon\nu$ is the circuit error rate as the expected number of errors in a circuit. In practice the approximation error $\mathcal{E}_F = \langle\psi_{id}|\rho_{err}|\psi_{id}\rangle$ is typically small and in the limiting case of white noise it decreases exponentially as $\mathcal{E}_F = 1/d$ due to $\rho_{err} = \mathrm{Id}/d$.

Assuming sufficiently deep, complex circuits, ref. [28] obtained an approximate bound for the commutator between the ideal and noisy quantum states as

$$\|[\rho_{id}, \rho]\|_1 \lessapprox \text{const} \times e^{-\xi}\xi/\sqrt{\nu}. \tag{10}$$

This bound confirms that as we increase the number of quantum gates ν in a circuit but keeping the circuit error rate ξ constant, the commutator norm decreases as $\propto 1/\sqrt{\nu}$ [28]. Furthermore, this function closely resembles to Eq. (1) which is a central aim of this work to explore.

C. White noise in random circuits

Random circuits have enabled quantum supremacy experiments using noisy quantum computers for two primary reasons: (a) the outputs of these circuits are hard to simulate classically and (b) they render local noise into global white noise [1], hence introducing only a trivial bias to the ideal probability distribution similarly as in Section I A.

Ref [23] considered random circuits consisting of s two-qubit gates, each of which undergoes two single qubit

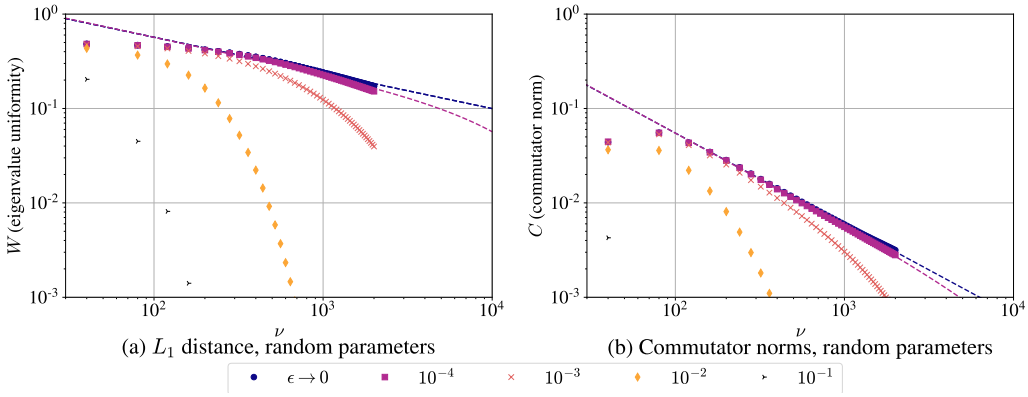


FIG. 1. Simulating families of 10-qubit Strong entangling layer (SEL) ansatz circuits [32] at random gate parameters for an increasing number ν of gates and per-gate depolarising error rates ϵ. (a) the uniformity measure $W(\nu)$ of the error eigenvalues of the density matrix from Eq. (12) closely match the theoretical model (dashed lines) for random circuits and confirm that increasing the number of gates in random circuits scrambles local noise into global white noise. (b) the commutator norm $C(\nu)$ from Eq. (14) is significantly smaller in absolute value and decreases with a larger polynomial degree (steeper slope of the dashed lines) than the uniformity measure – this suggests that the dominant eigenvector of the density matrix ρ approximately commutes with ρ even when noise is not well described by white noise. The $\epsilon \to 0$ simulations were approximated using $\epsilon = 10^{-8}$ ($\epsilon = 10^{-7}$) when calculating W (C).

(depolarising) errors each with probability $\tilde{\epsilon}$ (assuming single-qubit gates are noiseless). We can relate this to our model by identifying the local noise after each two-qubit gate with the error event in Eq. (7) via the probability $\epsilon = 1 - (1-\tilde{\epsilon})^2 = 2\tilde{\epsilon} - \tilde{\epsilon}^2$. Ref [23] then established the fidelity $\tilde{F}$ of the quantum state which one obtains from a noisy cross-entropy score as

$$\tilde{F} = e^{-2s\tilde{\epsilon} \pm O(s\tilde{\epsilon}^2)} = e^{-\xi \pm O(\epsilon\xi)}.$$

This coincides with our approximation from Eq. (9) up to an additive error in the exponent which, however, diminishes for low gate error rates. In the following we will thus assume $F \equiv \tilde{F}$.

Measuring these noisy states in a the standard measurement basis $\{|j\rangle\}_{j=1}^{d}$ produces a noisy probability distribution $\tilde{p}_{noisy}(j) = \langle j|\rho|j\rangle$. Ref. [23] established that this probability distribution rapidly approaches the white noise approximation $\tilde{p}_{wn} = Fp_{id} + (1-F)p_{unif}$. In particular, the total variation distance (via the l_1 norm $\|x\|_1 = \sum_i |x_i|$) between the two probability distributions is upper bounded as

$$\tfrac{1}{2}\|\tilde{p}_{noisy} - \tilde{p}_{wn}\|_1 \leq O(F\epsilon\sqrt{\nu}) = O(e^{-\xi}\xi/\sqrt{\nu}). \quad (11)$$

This expression is formally identical to the bound on the commutator norm in Eq. (10); Indeed if the noise in the quantum state approaches a white noise approximation, it implies that the commutator norm must also vanish in the same order.

On the other hand, the reverse is not necessarily true as Eq. (11) is a stronger condition than Eq. (10) as the latter only guarantees that the dominant eigenvector approaches $|\psi_1\rangle \approx |\psi_{id}\rangle$ but does not imply anything about the eigenvalue distribution of ρ or ρ_{err}.

III. NUMERICAL SIMULATIONS

A. Target metrics

In the NISQ-era comprehensive error correction will not be feasible and thus hope is primarily based on variational quantum algorithms [11–13, 33, 34]. In this paradigm a shallow, parametrised quantum circuit is used to prepare a parametrised quantum state that aims to approximate the solution to a given problem, typically the ground state of a problem Hamiltonian. Due to its shallow depth the ansatz circuit is believed to be error robust and its tractable parametrisation allows to explore the Hilbert space near the solution. On the other hand, such circuits are structurally quite different than random quantum circuits and it was already raised in ref. [23] whether error bounds on the white noise approximation extend to these shallow quantum circuits.

We simulate such quantum circuits under the effect of local depolarising noise – while note that a broad class of local coherent and incoherent error models can effectively be transformed into local depolarising noise using, e.g., twirling techniques or randomised compiling [35–38]. We analyse the resulting noisy density matrix ρ by calculating the following two quantities. First, we quantify 'closeness' to a white noise state from Eq. (2) by computing uniformity measure W as the l_1-distance between the uniform distribution and the non-dominant eigenvalues of the output state as

$$W := \frac{1}{2}\|p_{err} - p_{unif}\|_1 = \frac{1}{2}\sum_{k=2}^{d} \left|\frac{\lambda_k}{1-\lambda_1} - \frac{1}{d-1}\right|, \quad (12)$$

which only depends on spectral properties of the quan-

tum state and can thus be computed straightforwardly. We show in Statement 2 that W is proportional to the trace distance from a white noise quantum state as

$$\|\rho - \rho_{wn}\|_1 = (1-\lambda_1)W + \mathcal{E}_w, \tag{13}$$

uo to a bounded error $\mathcal{E}_w$. The uniformity measure W thus determines the bias in estimating any traceless expected value as discussed in Section I A.

Second, we calculate the commutator norm $\mathcal{E}_C$ from Section I A relative to $1-\lambda_1$ as

$$C := \frac{\|[\rho_{id}, \rho]\|_1}{1-\lambda_1} = \|[\rho_{id}, \rho_{err}]\|_1 + \mathcal{O}(\mathcal{E}_q), \tag{14}$$

which we relate to the commutator norm between the "error part" of the state ρ_{err} and the ideal quantum state ρ_{id} in Lemma 1. In the following, we will refer to C as the commutator norm – and recall that it determines the ultimate performance of purification-based error mitigation as discussed in Section I A.

B. Random states via Strong Entangling ansätze

We first consider a Strong Entangling ansatz (SEA): it is built of alternating layers of parametrised single-qubit rotations followed by a series of nearest-neighbour CNOT gates as illustrated in Fig. 5 – and assume a local depolarising noise with probability ϵ. We simulate random quantum circuits by randomly generating parameters $|\theta_k| \leq 2\pi$ – note that these circuits are not necessarily Haar-random distributed and thus results in Section II C do not necessarily apply.

We simulate 10-qubit circuits and in Fig. 1 (a) we plot the eigenvalue uniformity $W(\nu)$ while in Fig. 1 (b) we plot the commutator norm $C(\nu)$ for an increasing number ν of quantum gates – all datapoints are averages over ten random seeds. These results surprisingly well recover the expected behaviour of random quantum circuits as for small error rates $\epsilon \to 0$ both quantities $W(\nu)$ and $C(\nu)$ can be approximated by the function from Eq. (1) as we now discuss.

In Section II C we stated bounds of ref. [23] on the distance between $\tilde{p}_{noisy}$ and $\tilde{p}_{wn}$. Based on the assumption that these bounds also apply to the probability distributions $p_{noisy} = \langle\psi_k|\,\rho\,|\psi_k\rangle$ and $p_{wn} := \langle\psi_k|\,\rho_{wn}\,|\psi_k\rangle$ we derive in Statement 4 the approximate bound on the eigenvalue uniformity as

$$W = O\left(\frac{e^{-\xi}\xi/\sqrt{\nu}}{1-e^{-\xi}}\right).$$

Furthermore, by combining Eq. (14) and the bound in Eq. (10) we find that the commutator norm C is similarly bounded by the same function. On the other, Fig. 1 (b) suggests that the commutator norm decreases with a larger polynomial degree and thus we approximate both $W(\nu)$ and $C(\nu)$ using the function

$$f(\nu) = \alpha\frac{\xi e^{-\xi}}{\nu^\beta(1-e^{-\xi})} = \alpha/\nu^\beta + \mathcal{O}(\xi) \tag{15}$$

where we fit the two parameters α and β to our simulated dataset. The second equation above is an expansion for small circuit error rates ξ as detailed in Appendix A 2 a. Indeed, in Fig. 1 (blue circles) for small $\epsilon \to 0$ we observe a nearly linear function in the log-log plot in Fig. 1 and thus remarkably well recover the theoretical bounds with the polynomial power approaching $b \to 1/2$.

Furthermore, comparing Fig. 1 (b, blue circles) and Fig. 1 (a, blue circles) suggest that the commutator norm has both a significantly smaller absolute value (smaller α) and decreases at a faster polynomial rate (larger beta) than the uniformity measure. In fact, the commutator norm is more than two orders of magnitude smaller than the uniformity measure which suggests that even when ρ_{err} is not approximated well by a white noise state it, nevertheless, almost commutes with the ideal pure state ρ_{id}.

We finally consider how the absolute factor α depends on the number of qubits: we perform simulations at a small error rate $\epsilon \to 0$ and fit our model function $\alpha\nu^\beta$ to extract $\alpha(N)$ for an increasing number of qubits. The results are plotted in Fig. 7 (e) and suggest that the prefactor $\alpha(N)$ initially grows slowly but then saturates while note that a polylogarithmic depth is sufficient to reach anticoncentration [23].

C. Variational Hamiltonian Ansatz

Theoretical results guarantee that the SEL ansatz initialised at random parameters approach for an increasing depth unitary 2-designs thereby reproducing properties of random quantum circuits [39, 40]. It is thus not surprising that the model introduced in Section II C gives a remarkably good agreement between the SEL ansatz (dots on in Fig. 1) and genuine random circuits (fits as continuous lines in Fig. 1).

Here we consider the Hamiltonian Variational Ansatz (HVA) [41, 42] at more practical parameter settings: The HVA has the advantage that we can efficiently obtain parameters that increasingly better (as we increase the ansatz depth) approximate the ground state of a problem Hamiltonian – we will refer to these as VQE parameters. We also want to compare this circuit against random circuits and thus also simulate the HVA such that every gate receives a random parameter as detailed in Appendix B 1.

While the VQE parameter settings capture the relevant behaviour in practice as one approaches a solution, the random parameters are more relevant, e.g., at the early stages of a VQE parameter optimisation. Furthermore, as the circuit is entirely composed by Pauli terms in the problem Hamiltonian, the dimensionality of its dynamical Lie algebra is entirely determined by the problem Hamiltonian in contrast to an exponentially growing algebra of the SEL ansatz [24].

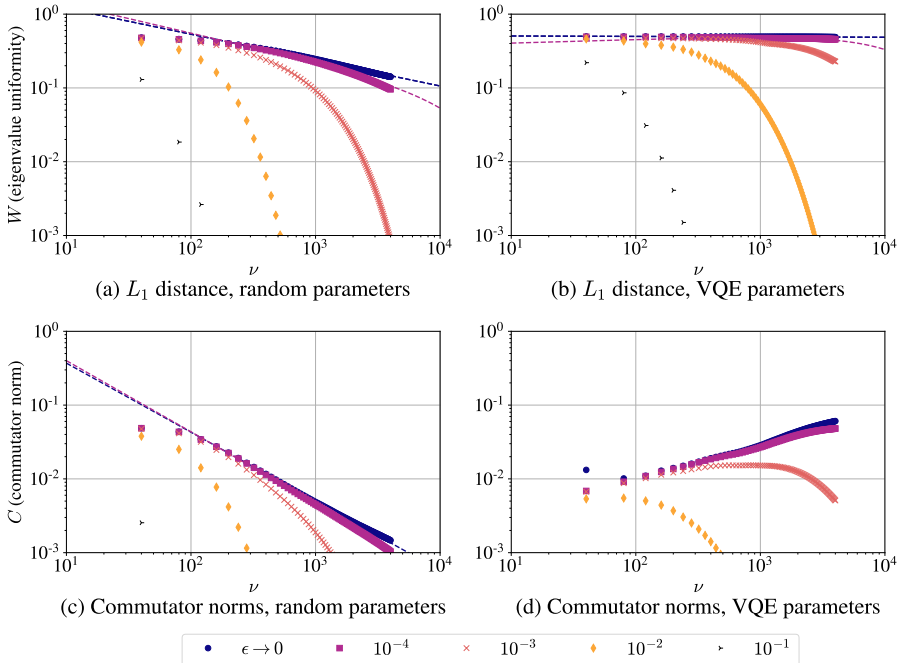


FIG. 2. **XXX Hamiltonian:** same simulations as in Fig. 1 but using 10-qubit HVA quantum circuits constructed for the XXX spin problem Hamiltonian. (a, c) at randomly chosen circuit parameters of the HVA we find the same conclusions as for random circuits in Fig. 1. (b) when the HVA circuit approximates the ground state of the Hamiltonian (VQE parameters) we find the noise in the circuit is not approximated well by white noise, i.e., the uniformity measure $W(\nu)$ is large and does not decrease as we increase ν. (d) On the other hand, the commutator norm $C(\nu)$ is significantly smaller than $W(\nu)$ confirming that the the ideal quantum state approximately commutes with the noisy one. The $\epsilon \to 0$ simulations were approximated using $\epsilon = 10^{-8}$ ($\epsilon = 10^{-7}$) when calculating W (C).

D. Heisenberg XXX spin model

We first consider a VQE problem of finding the ground state of the 1-dimensional XXX spin-chain model. We construct the HVA ansatz from Section III C for this problem Hamiltonian as a sum $\mathcal{H}_{\text{XXX}} = \mathcal{H}_0 + \mathcal{H}_1$ as

$$\mathcal{H}_0 = \sum_{k=1}^{N} \Delta_k Z_k,\ \mathcal{H}_1 = \sum_{k=1}^{N} [X_k X_{k+1} + Y_k Y_{k+1} + Z_k Z_{k+1}].$$

The Pauli operators XX, YY and ZZ determine couplings between nearest neighbour spins in a 1-D chain and we choose them to be of unit strength. Furthermore, Z_k are local on-site interactions $|\Delta_k| \leq 1$ that were generate uniformly randomly such that the Hamiltonian has a non-trivial ground state.

First, we simulate the HVA ansatz for $N = 10$ qubits with randomly generated circuit parameters as $|\theta_k| \leq 2\pi$ and plot results for an increasing number of quantum gates in Fig. 2 (a, c). We a find similar behaviour for the eigenvalue uniformity $W(\nu)$ as with random SEL circuits in Fig. 1 (a) and obtain a reasonably good fit for $\epsilon \to 0$ using our model function from Eq. (15). The commutator norm in Fig. 2 (c) is again significantly smaller in magnitude than the uniformity measure and decreases faster with a higher polynomial order similarly to as with the random SEL ansatz in Fig. 1 (b) .

Second, in Fig. 2 (b,d) we simulate the ansatz at the VQE parameters that approximate the ground state. Since the ansatz parameters become very small as one approaches an adiabatic evolution, it is not surprising that the output density matrix is not well-approximated by a white noise state: the uniformity measure is very large in Fig. 2 (b). The commutator norm in Fig. 2 (d) again, is significantly smaller than $W(\nu)$ and although it appears to slowly grow with ν, it appears to decrease for $\nu \to \infty$. This agrees with observations of ref. [28] that the circuits need not be random for the commutator to be sufficiently small in practice.

Furthermore, in Fig. 7 (a, b) we investigate the dependence on N and find that the prefactor α grows slowly and appears to saturate for $N \geq 10$.

E. TFI

In the next example we consider the transverse-field Ising (TFI) model $\mathcal{H}_{\text{TFI}} = \mathcal{H}_0 + \mathcal{H}_1$ using constant on-site interactions $h_i = 1$ and randomly generated coupling

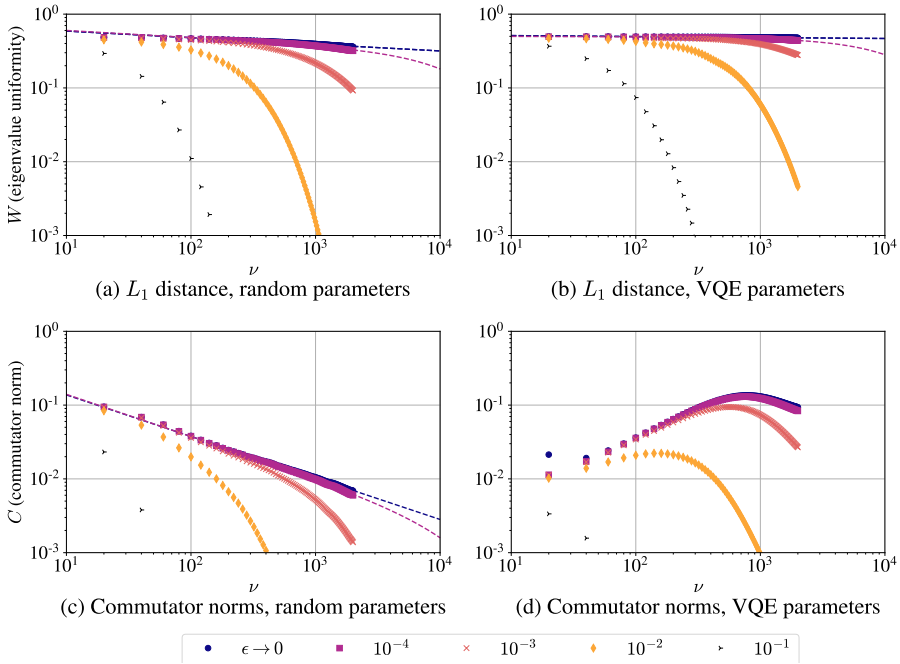


FIG. 3. **TFI** same simulations as in Fig. 1 but using 10-qubit HVA quantum circuits constructed for the TFI spin problem Hamiltonian. (a, c) at randomly chosen circuit parameters $W(\nu)$ decreases more slowly, in smaller polynomial order than random circuits – see text and see simulations with added layers of R_z gates in Fig. 6. (b) at the VQE parameters white noise is again not a good approximation, i.e., the uniformity measure $W(\nu)$ is large and does not decrease as we increase ν. (d) the commutator norm $C(\nu)$ is smaller than $W(\nu)$ in absolute value by an order of magnitude. The $\epsilon \to 0$ simulations were approximated using $\epsilon = 10^{-8}$ ($\epsilon = 10^{-7}$) when calculating W (C).

strengths $|J_i| \leq 1$ as

$$\mathcal{H}_0 = -\sum_i h_i X_i, \quad \mathcal{H}_1 = -\sum_i J_i Z_i Z_{i+1}. \tag{16}$$

We first simulate the HVA ansatz with random variational parameters in Fig. 3 (a, c). While at small error rates $\epsilon \to 0$ Fig. 3 (a, blue) can be fitted well with our polynomial approximation form Eq. (15), we observe that the eigenvalue uniformity $W(\nu)$ in Fig. 3 (a, blue) decreases with a small polynomial degree.

Indeed, as the HVA ansatz is specific to a particular Hamiltonian, its dynamical Lie algebra may have a low dimensionality [24] resulting in a limited ability to scramble local noise into white noise; this explains why in Fig. 3 (a) the uniformity measure decreases more slowly, i.e., in a smaller polynomial order, than random circuits. For this reason, we additionally simulate in Fig. 6 the TFI-HVA ansatz but with adding R_z gates in each layer whose generator is not contained in the problem Hamiltonian. The increased dimensionality of the dynamic Lie algebra, indeed, improves scrambling as the white noise approximation is clearly better in Fig. 6 – while note that the increased dimensionality may also lead to exponential inefficiencies in training the circuit [24].

In stark contrast to the case of the uniformity measure $W(\nu)$, we find that the commutator norm in Fig. 3 (c, blue) decreases substantially for an increasing ν despite the low dimensionality of the Lie algebra. This nicely demonstrates that a small commutator norm is a much more relaxed condition than white noise as the latter requires that the noise is fully scrambled in the entirety of the exponentially large Hilbert space. Finally, we simulate the TFI circuits at VQE parameters and find qualitatively the same behaviour as in the case of the XXX problem.

F. Quantum Chemistry: LiH

We consider a 6-qubit Lithium Hydride (LiH) Hamiltonian in the Jordan-Wigner encoding as a linear combination of non-local Pauli strings $P_k \in \{\mathrm{Id}, X, Y, Z\}^{\otimes N}$ as

$$\mathcal{H}_{LiH} = \sum_{k=1}^{r_h} h_k P_k. \tag{17}$$

We construct the HVA ansatz by splitting this Hamiltonian into two parts with $\mathcal{H}_0$ being composed of the diagonal Pauli terms in Eq. (17) while $\mathcal{H}_1$ composed of non-diagonal Pauli strings.

Such chemical Hamiltonians typically have a very large number of terms with $r_h \gg 1$ but a significant fraction only have small weights h_k thus the HVA would have

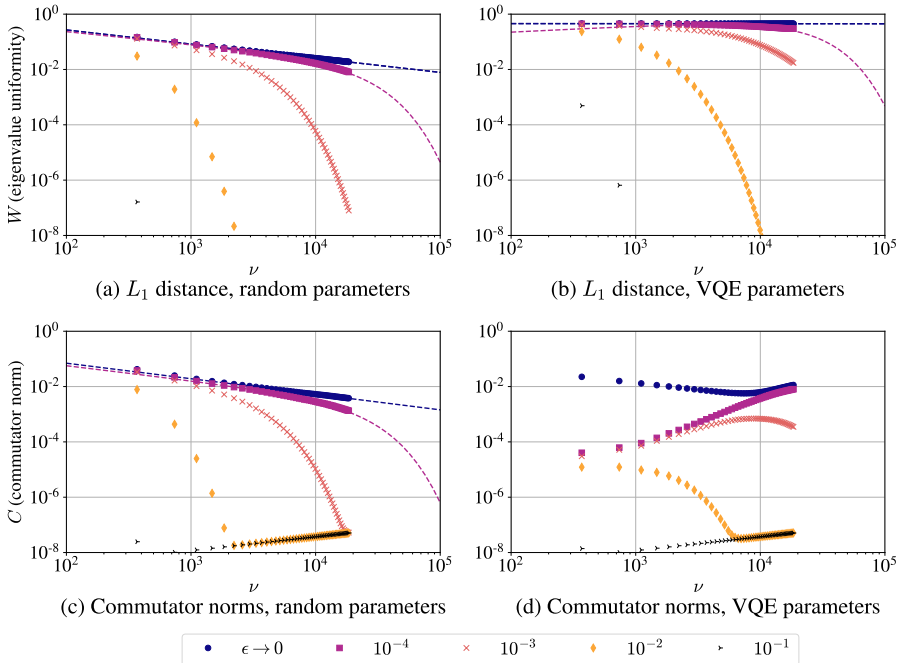


FIG. 4. **LiH** same simulations as in Fig. 1 but using 6-qubit HVA quantum circuits constructed for a LiH molecular Hamiltonian. (a, c) at randomly chosen circuit parameters both $W(\nu)$ and $C(\nu)$ decrease as expected for random circuits due our randomised compiling strategy [43, 44]. (b) at the VQE parameters white noise is an increasingly bad approximation, i.e., the uniformity measure $W(\nu)$ increases as we increase ν. (d) the commutator norm $C(\nu)$ is smaller than $W(\nu)$ in absolute value by 2 orders of magnitude. The $\epsilon \to 0$ simulations were approximated using $\epsilon = 10^{-8}$ ($\epsilon = 10^{-7}$) when calculating W (C).

a large number of gates with only very small rotation angles. For these reasons we construct a more efficient circuit whose basic building blocks are constructed using sparse compilation techniques [43, 44]: Each single layer in the HVA ansatz consists of gates that correspond to 100 randomly selected terms of the Hamiltonian with sampling probabilities $p_k \propto |h_k|$ proportional to the Pauli coefficients. This approach has the added benefit that it makes the circuit structure random as opposed to the fixed structures in Section III D and in Section III E.

Results shown in Fig. 4 (a,c) agree with our findings from the previous sections: at randomly chosen circuit parameters the uniformity measure decreases according to Eq. (15); the commutator norm similarly decreases but in a higher polynomial order while its absolute value is smaller by at least an order of magnitude. In contrast, Fig. 4 (b) suggests that the errors are not well approximated by white noise with a large and non-decreasing $W(\nu) \approx 0.5$. Furthermore, Fig. 4 (b) again confirms that despite white noise is not a good approximation, the commutator norm is small in absolute value, i.e., $\approx 10^{-3}$ in the practically relevant region. This guarantees a very good performance of the ESD/VD error mitigation techniques sufficient for nearly all practical purposes.

IV. DISCUSSION

Random quantum circuits—instrumental for demonstrating quantum advantage—are known to scramble local gate noise into global white noise for sufficiently long circuit depths [1]: general bounds have been proved on the approximation error which decrease as $\nu^{-1/2}$ as we increase the number ν of gates in the random circuit [23].

In this work we consider shallow-depth, variational quantum circuits that are typical in practical applications of near-term quantum computers and answer the question: can variational quantum circuits scramble local gate noise into global depolarising noise? While the answer to this question is relevant for the fundamental understanding of noise processes in near-term quantum devices, it has significant implications in practice: the degree to which local noise is scrambled into white noise determines the performance of a broad class of error mitigation techniques that are of key importance to achieving value with near-term devices [20]. As such, we derive two simple metrics that bound performance guarantees: first, the uniformity measure W characterises the performance of error mitigation techniques that assume global depolarising (white) noise [25]; second, the norm C of the commutator between the ideal and noisy quantum states determines the performance of purification-based error mitigation techniques [21, 22] via bounds of ref. [28].

We perform a comprehensive set of numerical experi-

ments to simulate typical applications of near-term quantum computers and analyse characteristics of noise based on the aforementioned two metrics. In all experiments in which we *randomly initialise parameters of the variational circuits* we semiquantitatively find the same conclusions. First, both metrics, the eigenvalue uniformity W and the commutator norm C are well described by our polynomial approximation from Eq. (15) for small gate error rates. Second, this confirms that, similarly to genuine random circuits, local errors get scrambled into global white noise with a polynomially decreasing approximation error as we increase the number of gates. Third, the commutator C decreases at a higher polynomial rate and has a significantly, by 1-2 orders of magnitude, smaller absolute value in the practically relevant region than the eigenvalue uniformity W. This confirms that purification based techniques are expected to have a superior performance compared to error mitigation techniques that, e.g., assume a global depolarising noise.

We then investigate the practically more relevant case when the ansatz circuits are initialised near the ground state of a problem Hamiltonian; in all cases we semiquantitatively find the same conclusions. First, the errors do not get scrambled into white noise and the approximation errors are large thus effectively prohibiting or at least significantly limiting the use of error mitigation techniques that assume global depolarising noise. Second, the commutator norm is quite small in absolute value, i.e., $\approx 10^{-2} - 10^{-4}$ in the practically relevant region; Since the ansatz circuit prepares the ground state, the square of the commutator norm determines the performance of ESD/VD thus for all applications we simulated we expect a very good performance of the ESD/VD approach. Third, we identify strategies to improve scrambling of local noise into global white noise as we increase circuit depth: We find that inserting additional gates to a HVA that is otherwise not contained in the problem Hamiltonian increases the dimensionality of the dynamic Lie algebra and thus leads to a reduction of both metrics. We find that applying randomised compiling to these non-random, practical circuits also reduces both metrics.

While purification-based techniques [21, 22] have been shown to perform well on specific examples, the present systematic analysis of circuit noise puts these results into perspective and demonstrates the following: First, the superior performance of the ESD/VD technique is not necessarily due to randomness in the quantum circuits – albeit, in deep and random circuits its performance is further improved. Second, while some error mitigation techniques perform well on quantum circuits well-described by white noise [25–27], we identify various practical scenarios where a limited performance is expected.

The present work advances our understanding of the nature of noise in near-term quantum computers and helps making progress towards achieving value with noisy quantum machines in practical applications. As such, results of the present work will be instrumental for identifying design principles that lead to robust, error-tolerant quantum circuits in practical applications.

Data availability

Numerical simulation code is openly available in the repository: github.com/jfold/shallow-circuit-noise.

ACKNOWLEDGEMENTS

The authors thank Simon Benjamin for his support. The authors thank Richard Meister for instructions on pyQuEST and Arthur Rattew for multiple discussions on quantum simulations. All simulations in this work were performed using the simulation tools QuEST [45] and its Python interface pyQuEST [46]. B.K. thanks the University of Oxford for a Glasstone Research Fellowship and Lady Margaret Hall, Oxford for a Research Fellowship. B.K. derived analytical results and contributed to writing the manuscript. J.F. was supported by the William Demant Foundation [grant number 18-4438]. J.F. performed numerical simulations and contributed to writing the manuscript.

Appendix A: Derivation of Eq. (6)

Recall that any quantum state can be transformed into a non-negative arrowhead matrix following Statement 1 from [28] as $\tilde{\rho} = F|\tilde{\psi}_{id}\rangle\langle\tilde{\psi}_{id}| + D + C$ with

$$\tilde{\rho} = \begin{pmatrix} F & C_2 & C_3 & \dots & C_d \\ C_2 & D_2 & & & \\ C_3 & & D_3 & & \\ \vdots & & & \ddots & \vdots \\ C_d & & & \dots & D_d \end{pmatrix}. \tag{A1}$$

We obtain the above matrix by applying a suitable unitary transformation $\tilde{\rho} := U\rho U^\dagger$ such that $|\tilde{\psi}_{id}\rangle := U|\psi_{id}\rangle = (1, 0, \dots 0)$ while $F, C_k, D_k \geq 0$ with $k \in \{2, 3, \dots, d\}$ with d denoting the dimension, and all other matrix entries are zero. Given the above arrowhead representation of a quantum state, one can analytically compute eigenvalues of the density matrix as roots of the

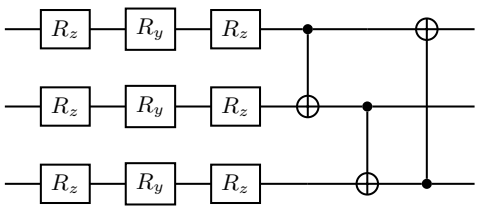


FIG. 5. A single layer of the Strong Entangling Layers ansatz for three qubits: it first applies single-qubit gates Ry, Rz and Ry on all qubits which is then followed by nearest neighbour CNOT gates.

following secular equation [28, 47]

$$P(x) = x - F + \sum_{k=2}^{d} \frac{C_k^2}{(D_k - x)} = 0. \tag{A2}$$

With this we compute the deviation between dominant eigenvalue λ_1 and the fidelity as

$$\begin{aligned}\lambda_1 - F =& \sum_{k=2}^{d} \frac{C_k^2}{(\lambda_1 - D_k)} \leq \max_k (\lambda_1 - D_k)^{-1} \sum_{k=2}^{d} C_k^2 \\ \leq& \|[\rho_{id}, \rho]\|^2 (2\lambda_1 - 1)^{-1},\end{aligned} \tag{A3}$$

where we have used that $D_k \leq \lambda_1$ and that all summands are non-negative as $D_k, C_k, \lambda_1 \geq 0$, and in the second inequality we have used the series of matrix norms $\sum_{k=2} C_k^2 = \|C\|_{HS}^2/2 = \|[\rho_{id}, \rho]\|_\infty^2$ as established in [28]. We have also introduced the abbreviation $\|[\rho_{id}, \rho]\|$ given all p-norms of the matrix $[\rho_{id}, \rho]$ are equivalent up to a constant factor. In particular, any p-norm of the commutator can be computed as $\|[\rho_{id}, \rho]\|_p = 2^{1/p}\sqrt{\mathrm{Var}[\rho]}$ where we used the quantum mechanical variance $\mathrm{Var}[\rho] := \langle \psi_{id}|\rho^2|\psi_{id}\rangle - F^2$ as established in [28]. Furthermore, in the second inequality in Eq. (A2) we have used that $\max_k(\lambda_1 - D_k)^{-1} = (\lambda_1 - D_2)^{-1} \leq (\lambda_1 - \lambda_2)^{-1} \leq (2\lambda_1 - 1)^{-1}$ by substituting the general inequality $\lambda_2 \leq (1 - \lambda_1)$ due to the fact that $\mathrm{tr}[\rho] = 1$.

By denoting the commutator norm as $\mathcal{E}_C$, we can thus finally conclude that $\lambda_1 - F \in O(\mathcal{E}_C)$ as stated in Eq. (6).

1. Trace distance from white noise states

In this section we evaluate analytically the trace distance of any quantum state ρ from the corresponding white noise state in Eq. (2) in terms of a distance between probability distributions.

Statement 1. *We can approximate the white noise-state in Eq. (2) in terms of the dominant eigengvalue λ_1 and the dominant eigenvector $|\psi_1\rangle$ of the quantum state as*

$$\rho_{wn} = \lambda\, |\psi_1\rangle\langle\psi_1| + (1 - \lambda_1)\mathrm{Id}/d + \mathcal{E}_w, \tag{A4}$$

up to an approximation error $\mathcal{E}_w$ that is bounded via Eq. (A6).

Proof. We start by approximating the weight η in Eq. (2) as $\eta \approx F \approx \lambda_1$ via Eq. (9) as well as we approximate the dominant eigenvalue using Eq. (6) and then collect the approximation errors as

$$\rho_{wn} = \lambda\, |\psi_{id}\rangle\langle\psi_{id}| + (1 - \lambda_1)\mathrm{Id}/d + \mathcal{E}_F + \mathcal{E}_C + O(\epsilon^2/\nu).$$

We now use results in [28] for bounding the distance between the ideal and noisy quantum states as

$$\begin{aligned}\||\psi_{id}\rangle\langle\psi_{id}| - |\psi_1\rangle\langle\psi_1|\|_1 =& \sqrt{1 - \langle\psi_{id}|\psi_1\rangle} \\ =& 1 - O\left(\frac{\mathcal{E}_C}{\lambda_1 - \lambda_2}\right),\end{aligned}$$

where $\mathcal{E}_C$ is the commutator norm from Eq. (6). We thus establish the approximation

$$\rho_{wn} = \lambda\, |\psi_1\rangle\langle\psi_1| + (1 - \lambda_1)\mathrm{Id}/d + \mathcal{E}_w, \tag{A5}$$

where we collect all approximation errors as

$$|\mathcal{E}_w| \leq |\mathcal{E}_F| + O(\epsilon^2/\nu) + O\left[\mathcal{E}_C(1 + \frac{1}{1 - \lambda_2/\lambda_1})\right]. \tag{A6}$$

□

Statement 2. *We define the eigenvalue uniformity as $W := \frac{1}{2}\|p_{err} - p_{unif}\|_1$ via the non-dominant eigenvalues of the density matrix $p_{err} := (\lambda_2, \lambda_3, \ldots, \lambda_d)/(1 - \lambda_1)$. This metric is related to the trace distance from a white noise state (as in Eq. (4)) as*

$$\|\rho - \rho_{wn}\|_1 = (1 - \lambda_1)W + \mathcal{E}_w, \tag{A7}$$

where the approximation error $\mathcal{E}_w$ is stated in Statement 1.

Proof. We substitute the approximation of ρ_{wn} from Eq. (A4) including the error term $\mathcal{E}_w$ and then we use the spectral decomposition of ρ to obtain the trace distance as

$$\begin{aligned}\|\rho - \rho_{wn}\|_1 =& \|\sum_{k=2}^{d} \lambda_k\, |\psi_k\rangle\langle\psi_k| - (1 - \lambda_1)\mathrm{Id}/d\|_1 + \mathcal{E}_w \\ =& \frac{1}{2}\sum_{k=2}^{d} |\lambda_k - \frac{1 - \lambda_1}{d}| + \mathcal{E}_w \qquad \text{(A8)} \\ =& \frac{1 - \lambda_1}{2}\|p_{err} - p_{unif}\|_1 + \mathcal{E}_w. \qquad \text{(A9)}\end{aligned}$$

In the second equation we analytically evaluated the trace distance and thus in the third equation we rewrite the result in terms of p_{err} which is our "error probability" distribution as $p_{err} := (\lambda_2, \lambda_3, \ldots, \lambda_d)/(1 - \lambda_1)$. □

Statement 3. *Alternatively to Statement 2, if a quantum state admits the decomposition in Eq. (8) then we can state the trace distance without approximation as*

$$\|\rho - \rho_{wn}\|_1 = \frac{(1 - \eta)}{2}\|p_\mu - p_{unif}\|_1. \tag{A10}$$

This is directly analogous to the uniformity measure of the non-dominant eigenvalues of ρ in Statement 2, however, this expression quantifies the uniformity of the probability distribution p_μ which are eigenvalues of the error density matrix ρ_{err}.

Let us assume the decomposition in Eq. (8). We find

the following result via a direct calculation as

$$\begin{aligned}\|\rho-\rho_{wn}\|_1 &= (1-\eta)\|\rho_{err}-\mathrm{Id}/d\|_1\\ &= (1-\eta)\|\sum_{k=1}^{d}\mu_k\,|\phi_k\rangle\langle\phi_k|-\mathrm{Id}/d\|_1\\ &= \frac{(1-\eta)}{2}\|\sum_{k=1}^{d}|\mu_k-1/d|\\ &= \frac{(1-\eta)}{2}\|p_\mu-p_{unif}\|_1\end{aligned}$$

where we have used the spectral resolution of the error density matrix and then analytically evaluated the trace distance. Given ρ_{err} is a positive-semidefinite matrix with unit trace, its eigenvalues μ_k form a probability distribution that we denote as p_μ.

2. Upper bounding the uniformity measure

In this section we upper bound the uniformity measure based on the number of gates and error rates in a quantum circuit.

Statement 4. *We adopt the bounds of [23] in Eq. (11) for the distance between probability distributions measured in the standard basis $\frac{1}{2}\|\tilde{p}_{noisy}-\tilde{p}_{wn}\|_1$ and assume the same bounds approximately apply to any measurement basis. Then, it follows that the uniformity measure from Statement 2 is approximately bounded by the same bounds as*

$$W = O(\frac{e^{-\xi}\xi/\sqrt{\nu}}{1-e^{-\xi}})+O(\frac{\mathcal{E}_w}{1-\lambda_1}),$$

where the approximation error $\mathcal{E}_w$ is stated in Statement 1.

Proof. Let us consider measurements performed in the basis as the eigenvectors of the density matrix which yield probabilities as the eigenvalues as

$$p_{noisy} = \langle\psi_k|\,\rho\,|\psi_k\rangle = (\lambda_1,\lambda_2\ldots,\lambda_d).$$

Measuring the white noise state in the same basis yields the following approximation of the probabilities using the error term from Eq. (A4) as

$$\begin{aligned}p_{wn} &:= \langle\psi_k|\,\rho_{wn}\,|\psi_k\rangle\\ &= (\lambda_1,\frac{1-\lambda_1}{d}\ldots,\frac{1-\lambda_1}{d})+\mathcal{E}_w.\end{aligned}$$

The distance of the above two measurement probability distributions is then

$$\frac{1}{2}\|p_{noisy}-p_{wn}\|_1 = (1-\lambda_1)W+\mathcal{E}_w,$$

where $W=\frac{1}{2}\|p_{err}-p_{unif}\|_1$ is our eigenvalue uniformity from Statement 2. Under the assumption that the upper bound on the measurement probabilities $\frac{1}{2}\|\tilde{p}_{noisy}-\tilde{p}_{wn}\|_1$ from Eq. (11) approximately holds for any measurement basis we can bound the eigenvalue uniformity as

$$\begin{aligned}W &= \frac{1}{2(1-\lambda_1)}\|p_{noisy}-p_{wn}\|_1+\frac{\mathcal{E}_w}{1-\lambda_1}\\ &\le O\left(\frac{F}{1-\lambda_1}\epsilon\sqrt{\nu}\right)+\frac{\mathcal{E}_w}{1-\lambda_1}\\ &= O\left(\frac{e^{-\xi}\xi/\sqrt{\nu}}{1-e^{-\xi}}\right)+O\left(\frac{\mathcal{E}_w}{1-\lambda_1}\right).\end{aligned}$$

In the last equation we introduced the approximation of F from Eq. (9) as well as the approximate dominant eigenvalue from Eq. (6). □

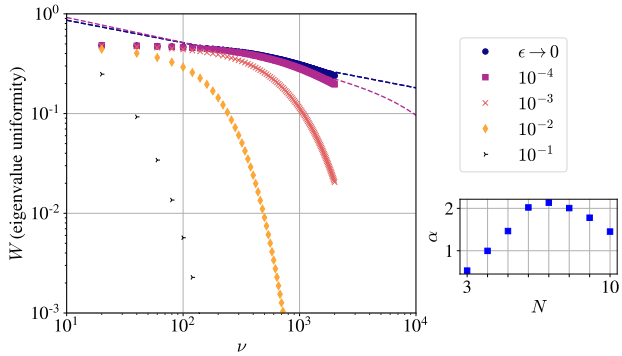


FIG. 6. (left) TFI-HVA ansatz: same simulations as in Fig. 3 (a) but with added parametrised R_z gates after each layer. The additional gates increase the dimensionality of the dynamic Lie algebra which leads to a faster scrambling of local gate noise into white noise, e.g., the $\epsilon\to 0$ curve is steeper than in Fig. 3 (a). See Appendix B for more details. (right) the dependence on the number of qubits shows a very similar trend as without the Rz gates, i.e., compare to Fig. 7 (c).

a. *Expanding the upper bound*

We now expand the upper bound from Statement 4 for small ξ as. More specifically, we consider the parametrised fit function from Eq. (15) and substitute the Taylor expansion $e^{-\xi}=1-\xi+\xi^2+\ldots$ as

$$\begin{aligned}\alpha\frac{e^{-\xi}\xi/\sqrt{\nu}^{\beta}}{1-e^{-\xi}} &= \alpha\frac{e^{-\xi}}{\nu^\beta}\frac{\xi}{\xi-\xi^2/2+\ldots}\\ &= \alpha\frac{e^{-\xi}}{\nu^\beta}\frac{1}{1-\xi/2+\ldots}\\ &= \alpha\frac{1}{\nu^\beta}\frac{1-\xi+\ldots}{1-\xi/2+\ldots}\\ &= \frac{\alpha}{\nu^\beta}+O(\xi).\end{aligned}$$

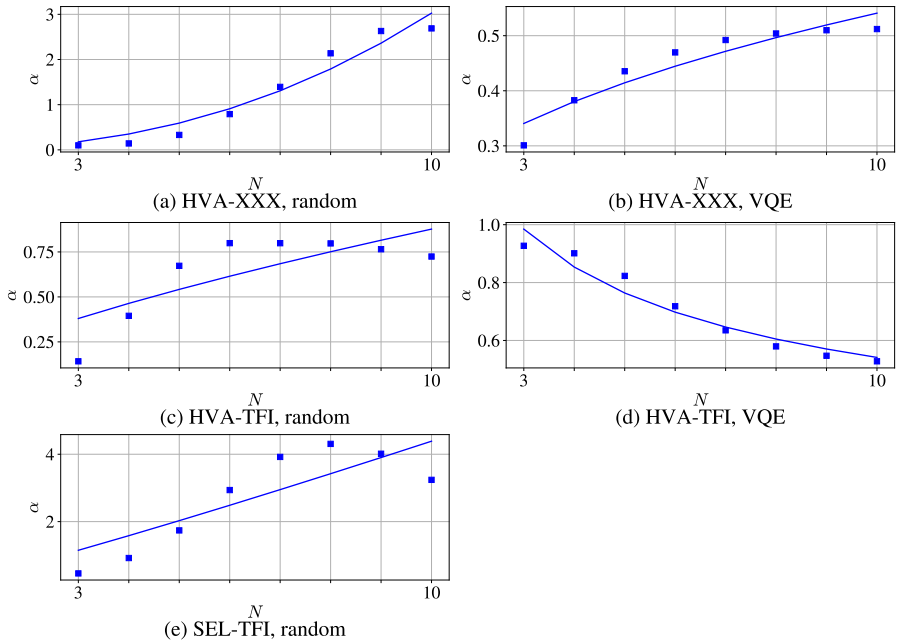


FIG. 7. Fit parameters α from Eq. (15) for an increasing number of qubits: The circuits in Figs. 1 to 4 at $\epsilon \to 0$ were simulated for an increasing number of qubits and the curve from Eq. (15) was fitted.

3. Commutator norm

Lemma 1. *The commutators norms are approximately related as*

$$\frac{\|[\rho_{id}, \rho]\|_1}{1-\lambda_1} = \|[\rho_{id}, \rho_{err}]\|_1 + \mathcal{E}_q, \tag{A11}$$

up to the approximation error $\mathcal{E}_q$.

Proof. Using the decomposition from Eq. (8) we obtain

$$\begin{aligned}\|[\rho_{id}, \rho]\|_1 &= \|[\rho_{id}, \eta\rho_{id}] + [\rho_{id}, (1-\eta)\rho_{err}]\|_1 \\ &= (1-\eta)\|[\rho_{id}, \rho_{err}]\|_1\end{aligned}$$

We can approximate $\eta = \lambda_1 + \mathcal{O}(\mathcal{E}_F) + \mathcal{O}(\mathcal{E}_C)$ via Eq. (9) and Eq. (6) and obtain that

$$\frac{\|[\rho_{id}, \rho]\|_1}{1-\lambda_1} = \|[\rho_{id}, \rho_{err}]\|_1 + \mathcal{E}_q. \tag{A12}$$

The error term can be obtained via the triangle inequality $|\mathcal{E}_q| \leq [\mathcal{O}(\mathcal{E}_F\mathcal{E}_C) + \mathcal{O}(\mathcal{E}_C^2)]/(1-\lambda_1)$. □

Appendix B: Further details of numerical simulations

1. The SEL and HVA ansätze

The circuit structure of the SEL ansatz used in Fig. 1 is illustrated in Fig. 5: it consists of alternating layers of parametrised single-qubit rotations and a ladder of nearest-neighbour CNOT gates.

Let us now define the HVA ansatz. In particular, recall that the HVA ansatz is a discretisation of the adiabatic evolution

$$U(\underline{\beta}, \underline{\gamma}) = \prod_{k=1}^{\nu} e^{-i\gamma_k \mathcal{H}_1} e^{-i\beta_k \mathcal{H}_0},$$

which is applied to the initial state as the ground state of the trivial Hamiltonian $\mathcal{H}_0$.

The individual evolutions are then trotterised such that a piece of time evolution $e^{-i\gamma_k \mathcal{H}_1}$ is broken up into products of evolution operators under the individual Hamiltonian terms as

$$e^{-i\gamma_k \mathcal{H}_1} \to \prod_{l=1}^{r_h} e^{-i\gamma_k h_l P_l}.$$

Above we utilised the decomposition of the non-trivial part of the Hamiltonian $\mathcal{H}_1 = \sum_{l=1}^{r_h} h_l P_l$ into Pauli strings $P_l \in \{\mathrm{Id}, X, Y, Z\}^{\otimes N}$.

We set the circuit parameter as $\gamma_k = k/\nu$ and $\beta_k = 1 - k/\nu$, such that the circuit approximates a discretised adiabatic evolution between $\mathcal{H}_0$ and $\mathcal{H}_1$ – and we will refer to these as VQE parameters.

In the case of random parametrisation of the HVA ansatz, every gate implementing the evolution under a single Pauli string $e^{-i\gamma_k h_l P_l}$ is assigned a random parameter as $e^{-i\theta_q P_l}$ with $|\theta_q| \leq 2\pi$.

2. Inserting additional gates to the TFI ansatz

In Fig. 6 we repeated the same simulation as in Fig. 3 (a), i.e., using a HVA ansatz for the TFI spin model at random circuit parameters, but we appended to each layer a series of parametrised Rz gates on each qubit. This guarantees that the dynamic Lie algebra generated by the Pauli terms of the TFI problem in Eq. (16) is expanded by the inclusion of Pauli Z operators. Increasing the circuit depth of the HVA ansatz thus leads to a faster increase of the dimensionality of the Lie algebra which demonstrably leads to a faster scrambling of local noise into global white noise, e.g., steeper slope of the $\epsilon \to 0$ fit in Fig. 6 than in Fig. 3.

3. Scaling with the number of qubits

In Fig. 7 we simulate the same circuits as in Figs. 1 to 4 at error rates $\epsilon \to 0$ and plot the fit parameter α—which is the prefactor in Eq. (15)—for an increasing number of qubits. The results appear to confirm an asymptotically non-increasing trend confirming theoretical expectations of [23] for random circuits whereby α is constant bounded in terms of the number of qubits.

[1] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, *et al.*, Quantum supremacy using a programmable superconducting processor, Nature **574**, 505 (2019).

[2] M. Tillmann, B. Dakić, R. Heilmann, S. Nolte, A. Szameit, and P. Walther, Experimental boson sampling, Nature photonics **7**, 540 (2013).

[3] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature **549**, 195 (2017).

[4] D. Jafferis, A. Zlokapa, J. D. Lykken, D. K. Kolchmeyer, S. I. Davis, N. Lauk, H. Neven, and M. Spiropulu, Traversable wormhole dynamics on a quantum processor, Nature **612**, 51 (2022).

[5] C. Kokail, C. Maier, R. van Bijnen, T. Brydges, M. K. Joshi, P. Jurcevic, C. A. Muschik, P. Silvi, R. Blatt, C. F. Roos, and P. Zoller, Self-verifying variational quantum simulation of lattice models, Nature **569**, 355 (2019).

[6] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. Sawaya, *et al.*, Quantum chemistry in the age of quantum computing, Chemical reviews **119**, 10856 (2019).

[7] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, Quantum computational chemistry, Reviews of Modern Physics **92**, 015003 (2020).

[8] B. Bauer, S. Bravyi, M. Motta, and G. K.-L. Chan, Quantum algorithms for quantum chemistry and quantum materials science, Chemical Reviews **120**, 12685 (2020).

[9] M. Motta and J. E. Rice, Emerging quantum computing algorithms for quantum chemistry, WIREs Computational Molecular Science **12**, e1580 (2022).

[10] J. Preskill, Quantum computing in the nisq era and beyond, Quantum **2**, 79 (2018).

[11] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, *et al.*, Variational quantum algorithms, Nature Reviews Physics **3**, 625 (2021).

[12] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan, Hybrid quantum-classical algorithms and quantum error mitigation, Journal of the Physical Society of Japan **90**, 032001 (2021).

[13] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W.-K. Mok, S. Sim, L.-C. Kwek, and A. Aspuru-Guzik, Noisy intermediate-scale quantum algorithms, Rev. Mod. Phys. **94**, 015004 (2022).

[14] M. Cerezo, G. Verdon, H.-Y. Huang, L. Cincio, and P. J. Coles, Challenges and opportunities in quantum machine learning, Nature Computational Science **2**, 567 (2022).

[15] B. van Straaten and B. Koczor, Measurement cost of metric-aware variational quantum algorithms, PRX Quantum **2**, 030324 (2021).

[16] B. Koczor and S. C. Benjamin, Quantum analytic descent, Phys. Rev. Res. **4**, 023017 (2022).

[17] H.-Y. Huang, R. Kueng, and J. Preskill, Predicting many properties of a quantum system from very few measurements, Nature Physics **16**, 1050 (2020).

[18] G. Boyd and B. Koczor, Training variational quantum circuits with covar: Covariance root finding with classical shadows, Phys. Rev. X **12**, 041022 (2022).

[19] H. H. S. Chan, R. Meister, M. L. Goh, and B. Koczor, Algorithmic shadow spectroscopy, arXiv preprint arXiv:2212.11036 (2022).

[20] Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O'Brien, Quantum error mitigation, arXiv preprint arXiv:2210.00921 (2022).

[21] B. Koczor, Exponential Error Suppression for Near-Term Quantum Devices, Phys. Rev. X **11**, 031057 (2021).

[22] W. J. Huggins, S. McArdle, T. E. O'Brien, J. Lee, N. C. Rubin, S. Boixo, K. B. Whaley, R. Babbush, and J. R. McClean, Virtual distillation for quantum error mitigation, Phys. Rev. X **11**, 041036 (2021).

[23] A. M. Dalzell, N. Hunter-Jones, and F. G. Brandão, Random quantum circuits transform local noise into global white noise, arXiv preprint arXiv:2111.14907 (2021).

[24] M. Larocca, P. Czarnik, K. Sharma, G. Muraleedharan, P. J. Coles, and M. Cerezo, Diagnosing barren plateaus with tools from quantum optimal control, Quantum **6**, 824 (2022).

[25] J. Vovrosh, K. E. Khosla, S. Greenaway, C. Self, M. S. Kim, and J. Knolle, Simple mitigation of global depolarizing errors in quantum simulations, Phys. Rev. E **104**, 035309 (2021).

[26] S. Endo, S. C. Benjamin, and Y. Li, Practical quantum error mitigation for near-future applications, Phys. Rev. X **8**, 031027 (2018).

[27] A. Strikis, D. Qin, Y. Chen, S. C. Benjamin, and Y. Li, Learning-based quantum error mitigation, PRX Quantum **2**, 040330 (2021).

[28] B. Koczor, The dominant eigenvector of a noisy quantum state, New Journal of Physics **23**, 123047 (2021).

[29] B. Koczor and S. C. Benjamin, Quantum natural gradient generalized to noisy and nonunitary circuits, Phys. Rev. A **106**, 062416 (2022).

[30] T. E. O'Brien, G. Anselmetti, F. Gkritsis, V. Elfving, S. Polla, W. J. Huggins, O. Oumarou, K. Kechedzhi, D. Abanin, R. Acharya, *et al.*, Purification-based quantum error mitigation of pair-correlated electron simulations, arXiv preprint arXiv:2210.10799 (2022).

[31] H. Jnane, B. Undseth, Z. Cai, S. C. Benjamin, and B. Koczor, Multicore Quantum Computing, Phys. Rev. Appl. **18**, 044064 (2022).

[32] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, Circuit-centric quantum classifiers, Physical Review A **101**, 032308 (2020).

[33] B. Koczor, S. Endo, T. Jones, Y. Matsuzaki, and S. C. Benjamin, Variational-state quantum metrology, New J. Phys. **22**, 083038 (2020).

[34] J. Foldager, A. Pesah, and L. K. Hansen, Noise-assisted variational quantum thermalization, Scientific reports **12**, 1 (2022).

[35] M. Silva, E. Magesan, D. W. Kribs, and J. Emerson, Scalable protocol for identification of correctable codes, Phys. Rev. A **78**, 012347 (2008).

[36] E. Magesan, J. M. Gambetta, and J. Emerson, Characterizing quantum gates via randomized benchmarking, Phys. Rev. A **85**, 042311 (2012).

[37] Z. Cai and S. C. Benjamin, Constructing smaller pauli twirling sets for arbitrary error channels, Sci. Rep. **9**, 1 (2019).

[38] Z. Cai, X. Xu, and S. C. Benjamin, Mitigating coherent noise using Pauli conjugation, npj Quantum Info. **6**, 1 (2020).

[39] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, Nature communications **9**, 1 (2018).

[40] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, Cost function dependent barren plateaus in shallow parametrized quantum circuits, Nature communications **12**, 1 (2021).

[41] D. Wecker, M. B. Hastings, and M. Troyer, Progress towards practical quantum variational algorithms, Physical Review A **92**, 042303 (2015).

[42] R. Wiersema, C. Zhou, Y. de Sereville, J. F. Carrasquilla, Y. B. Kim, and H. Yuen, Exploring entanglement and optimization within the hamiltonian variational ansatz, PRX Quantum **1**, 020319 (2020).

[43] E. Campbell, Random compiler for fast hamiltonian simulation, Phys. Rev. Lett. **123**, 10.1103/physrevlett.123.070503 (2019).

[44] Y. Ouyang, D. R. White, and E. T. Campbell, Compilation by stochastic Hamiltonian sparsification, Quantum **4**, 235 (2020).

[45] T. Jones, A. Brown, I. Bush, and S. C. Benjamin, QuEST and high performance simulation of quantum computers, Scientific reports **9**, 1 (2019).

[46] R. Meister, pyquest - a python interface for the quantum exact simulation toolkit (2022).

[47] D. O'leary and G. Stewart, Computing the eigenvalues and eigenvectors of symmetric arrowhead matrices, Journal of Computational Physics **90**, 497 (1990).

Appendix C

Paper C (preprint version)

THE FOLLOWING PAGES CONTAIN A COPY OF THE PREPRINT VERSION OF THE PAPER

Actively Learning Quantum Machine Learning Architectures from Related Problems

Jonathan Foldager*
Cognitive Systems, DTU Compute, Technical University of Denmark
(Dated: April 14, 2023)

With near-term quantum machine learning we can train variational quantum algorithms to perform hard tasks such as groundstate estimation of Hamiltonians. Such circuits typically involve several design choices, with a number of hyperparameters. In this paper, a very general framework using classical algorithmic agency is introduced. Based on a combination of Active Learning and Bayesian Optimization to guide experimental design of quantum models the approach aims to provide better initialization strategies for unseen cost functions. The numerical experiments focuses on the variational quantum eigensolver and results reveal that the agency outperforms several search strategies and thus suggests that lessons learned on one problem Hamiltonian can be used for another one. Results for different quantum Hamiltonians are presented and a discussion various agent strategies is provided.

I. INTRODUCTION

The field of quantum machine learning (QML) combines quantum computing and machine learning for a multitude of tasks [1, 2]. Many of the so-called first-wave QML proposals require capabilities, such as quantum random access memory (QRAM), that current and arguably near-future quantum technology cannot provide [3], and even if it could there are still unknowns with respect to applicability[4]. Instead, periodic optimism and a lot of research has been directed at the noisy intermediate-scale quantum (NISQ) technology and many ideas on how to leverage the quantum power here have been proposed [5, 6]. Problems involving learning a quantum circuit that prepares a quantum state of interest, such as the groundstate of a quantum Hamiltonian, have been argued to be a meaningful subject for NISQ machines and a potential area for early stage quantum advantage. Some of the first ideas realizing this include the variational quantum eigensolver (VQE) [7] and the quantum approximate optimization algorithm (QAOA) [8], which both are instances of variational quantum algorithms (VQAs); protocols which aims at learning a low-depth quantum circuit preparing useful quantum states.

The principle of VQAs is to parameterize the quantum processor itself, that is, parameterize the quantum gates (or more general quantum channels [9]) themselves, define a loss function and subsequently combine the quantum device with a classical computer which learns these parameters by minimizing the loss with respect to the loss function. For a gate-model quantum computer, this means that each gate is assigned a real number θ_i that controls *how* the corresponding gate operates. Using samples obtained from preparing and observing the output state multiple times and a classical optimizer, the parameters $\boldsymbol{\theta} \in \mathbb{R}^P$ where $[\boldsymbol{\theta}]_i = \theta_i$ are learned such that the quantum computer prepares a quantum state of interest. Formally, if $U(\boldsymbol{\theta})$ represents the parameterized quantum circuit unitary acting on a reference state $|\underline{0}\rangle$, the produced state becomes $U(\boldsymbol{\theta})\,|\underline{0}\rangle = |\psi(\boldsymbol{\theta})\rangle$. The goal of VQE is to learn the parameters

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \langle(\boldsymbol{\theta})|\,\mathcal{H}\,|\psi(\boldsymbol{\theta})\rangle \tag{1}$$

for a given Hamiltonian operator $\mathcal{H}$, which we shall assume is a parameterized sum of Pauli tensor products

$$\begin{aligned}\mathcal{H} := \mathcal{H}(\mathbf{J},\mathbf{b}) = &\sum_{(\alpha_i,\alpha_j)} \sum_i J_{ij}^{(\alpha_i,\alpha_j)} \sigma_i^{\alpha_i} \sigma_{i+1}^{\alpha_j} \\ &+ \sum_{\alpha_i} \sum_i b_i^{\alpha_i} \sigma_i^{\alpha_i}\end{aligned} \tag{2}$$

where $\mathbf{J}$ houses all interaction terms, $\mathbf{b}$ contains all bias terms and $\alpha_i \in \{X, Y, Z\}$ are local Pauli observables acting on qubit i. In this work we only consider spin ring systems, hence it is sufficient to sum over neighboring spins in eq. (2). Research has gone into investigating VQAs for variety of tasks beyond ground state preparations [10–13], and some have even considered these circuits themselves as generative machine learning models [14]. Applications of VQAs beyond quantum physics have been proposed in other scientific areas such as drug-discovery, material and molecular science, machine learning and optimization . Although the method and result in this paper easily can be generalized to other applications with VQAs, we shall focus on the task of VQE not only because the problem itself is interesting but also because many problems can be reformulated into a VQE-like optimization problem.

A. Motivation

When designing the variational algorithm, a number of ad hoc decisions are typically made: picking a cost function, ansatz strategy (overall architecture of quantum circuit, including circuit depth, number of ancillary qubits, etc.), parameter initialization, classical optimizer etc. These are so-called *hyperparameters* and can have

* Correspondence:jonf@dtu.dk

considerable impact on expressiveness, learnability and convergence speed [15]. This is also well known in classical machine learning, where hyperparameters often are chosen based on grid-search. Such strategy can a reasonable choice for small search spaces but is tedious and scales exponentially in the number of hyperparameters. Furthermore, and even more relevant for problems in physics, it does not seek to exploit potential knowledge on other perhaps very similar problems (for VQE this could be slightly different Hamiltonians). One proposed method to find hyperparameters in an automatic way is Bayesian optimization (BO)[16]. BO is particularly well-suited when 1) we can evaluate the loss objective, but we do not have access to the gradients, and 2) we can only allow a fairly small amount of samples e.g. due to costs. Such is exactly the case for hyperparameters in a VQA's (or classically in neural networks), where we wish to find the global optimum of a "black-box" function $f(x)$ with respect to input x while having no closed-form (and therefore no gradients) of f w.r.t. x. In such problems, we are allowed to query the function for specific x_i, and these samples might be noisy:

$$y_i = f(x_i) + \epsilon \tag{3}$$

where ϵ is noise term often assumed to follow a normal distribution. However, in this paper we are not only interested in BO performed for a particular problem. In fact, we want to learn about the relationship

$$\text{(hyperparameters,problem)} \rightarrow \text{loss}$$

with focus on hyperparameters of VQEs for various problems (Hamiltonians), hence our loss is the corresponding achieved minimum energy estimate. With a near-term quantum computer, we can perform VQE on a variety of Hamiltonians and it might be of interest to an experimentalist wanting to learn how to design VQA in automatic fashion across multiple Hamiltonians. The underlying assumption is that, tiny changes in the Hamiltonian coefficients not only yields tiny changes in the corresponding groundstate but also in optimal choice of hyperparameters. We do not necessarily expect correspondingly tiny changes in the optimal circuit parameters. A strategy to choose which Hamiltonians to simulate in order to learn as much as possible about the above relationship in as few experiments as possible is therefore needed. For this purpose active learning (AL) can be used. AL is very similar to BO except the way to query new datapoints. As opposed to BO, where the goal is to find a global minimum, AL seeks to learn the underlying loss landscape. If a BO strategy of minimizing a Hamiltonian expectation were to pick the coefficients of that same Hamiltonian it would likely try out large numerical Hamiltonian coefficients, which is not of particular interest or meaningfulness. However, AL would choose Hamiltonian coefficients that maximize how much we learn about the aforementioned relationship. We elaborate on differences in query strategies in section II B and section II C.

In this paper, we propose a simple but general protocol that combines AL and BO to learn better initialization and learning strategies for unseen problems. We call this *classical algorithm agency* inspired by the classical machine learning literature. Applications of this method include scenarios such as having a collection of Hamiltonians and wanting to automate experimental design by learning across problems. Another application could also be learning the expectation value of an operator for various inter-atomic distances/interactions expressed in the Hamiltonian. In other words, we "reformulate" the VQE as a quantum machine learning problem where VQE for one specific Hamiltonian becomes one datapoint consisting of hyperparameters and Hamiltonian coefficients. Instead of sampling randomly in this huge space, we employ AL to iteratively select the next Hamiltonian coefficients and subsequently perform BO. As a consequence, we only have to specify the search space $\mathcal{S}$ and search time t. Our approach illustrated in fig. 1 (right) and compared to normal settings (left). Indeed this protocol is very general and incorporating domain knowledge into the search space+time can be beneficial. We note that while our framework is very general, we only investigate the specific case of VQE. Furthermore, we test our algorithmic agent up against both random- and grid search for spin-rings, and we discuss future work to compare to the zoo of approaches in the VQA field. We illustrate the framework in fig. 1.

B. Related work

An approach similar to ours, called Meta-VQE [17] was recently published, and hence we adapted this name to our procedure. Here the authors propose to use an encoding layer in their quantum circuit followed by a processing layer, the latter being the normal VQE part. The encoding layer should contain information about the problem Hamiltonian coefficients $\boldsymbol{C}$ and thus their proposed ansatz is given by

$$|\psi\rangle = \mathcal{U}_p(\boldsymbol{\theta}_p)\mathcal{U}_e(\boldsymbol{C}, \boldsymbol{\theta}_e)\,|0\rangle \tag{4}$$

where subscript p and e refers to processing and encoding layers, respectively. Our method is different from Ref. [17] in five important aspects. First, we don't have an encoding quantum circuit layer, but instead, we hand the task of learning this Hamiltonian manifold to our classical agent. Hence we use less quantum resources. Second, the Hamiltonians (training points) our method chooses is automatically, iteratively selected by the agent and not by a human. Third, we incorporate the circuit architecture as part of the agent decision process using Bayesian Optimization compared to having a static circuit hyperparameters. Fourth, one of the goals of Ref. [17] was to be able to predict the ground-state energy of the Hamiltonian within a certain trust region. Our method also provides this opportunity, however, it not only comes with a prediction, but also an uncertainty estimate on that

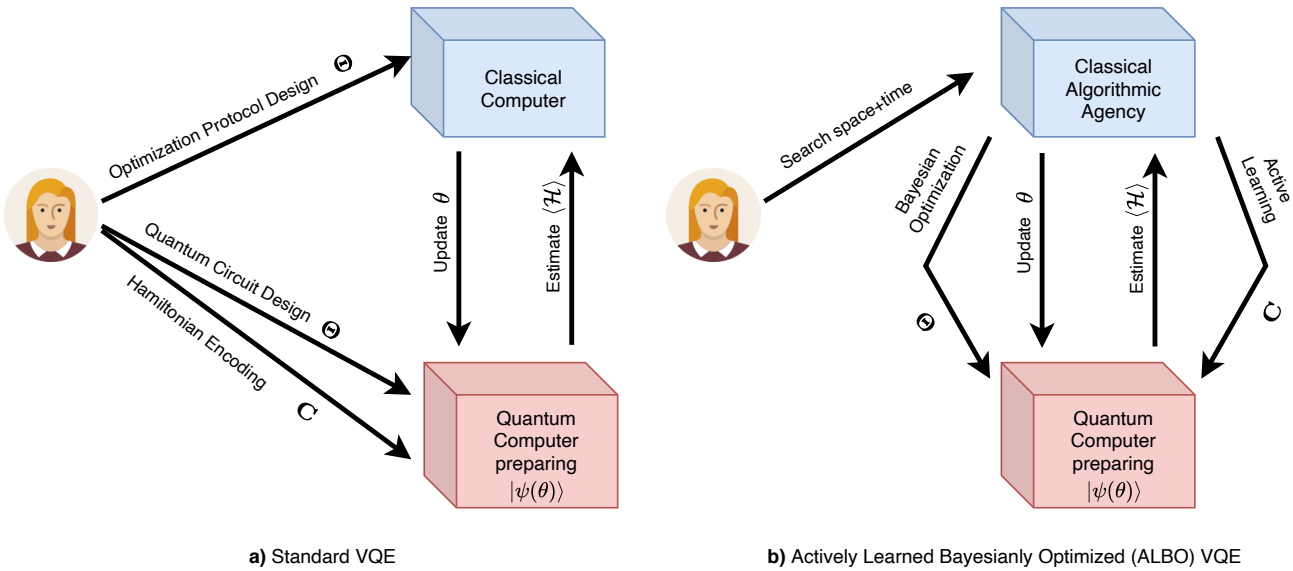


Figure 1: a) normal VQE protocol where the scientist designs the quantum circuit and optimization protocol with hyperparameters $\boldsymbol{\Theta}$ for optimizing the expectation of a single Hamiltonian with coefficients $\boldsymbol{C}$. **b)** actively learned and Bayesianly optimized (ALBO) VQA protocol where a search space for the hyperparameters are defined by the scientist, but the optimal hyperparameters for each specific Hamiltonian with coefficients $\boldsymbol{C}$ are found using Bayesian Optimization and multiple Hamiltonians are investigated with Active Learning.

prediction which can be useful to guide whether or not to make the experiment or trust the prediction. Lastly, we benchmark not only the final state fidelity with the true groundstate, but also the *regret* which is a measure of both fidelity and how convergence speed.

There are many complex aspects of finding an appropriate VQA archetectures to a given problem such as cost function, avoiding traps and potential BPs [18], maximizing expressibility, hardware efficiency, error-mitigation strategy while still keeping a low-depth to avoid error accumulations. Some of these challenges can be overcome by use of different ansatz classes such as the quantum convolutional neural networks[19] or exploiting problem symmetries [20]. Better parameter initialization circuit parameters of have also been proposed to minimize barren plateaus [21, 22]. Interesting theoretical insights into parameter concentrations was also been introduced [23]. Ref. [15] studied the general expressiveness of quantum circuits by introducing a expressiveness metric based on the KL-divergence and subsequently comparing various VQA architectures. Other approaches implement adaptive ansätze [24, 25], which can be thought of as a reinforcement-like strategy. More general approaches to circuit design have been proposed [26, 27] as well as automated searches [28] and meta-learning the parameters themselves [29, 30]. Interesting work has also been put into differentiable circuit design [31] as well as using a combination of many classical machine learning approaches to learn the best architecture [32]. Using BO with VQAs has been done before including parameter initialization [33], as a set in hyperparameter tuning [32], and using BO to train the parameters instead of gradient-based methods [34, 35]. We did not find any literature on using active learning with quantum circuits, however, active learning has been used before in learning to create quantum experiments [36].

II. BACKGROUND

In this section, we introduce variational quantum algorithms (VQAs) followed by a review of Bayesian Optimization (BO) and Active Learning (AL). We then explain how these two are combined into the classical algorithmic agent performing meta-VQE, which we call *ALBO*. Finally, we remark the details of the numerical simulations constituting the results section.

A. Preliminaries

We investigate a pure qubit system consisting of N qubits, which is represented by a complex vector $|\psi\rangle \in \mathbb{C}^{2^N}$ in a $d = 2^N$ dimensional Hilbert space. The qubits can be in a superposition of the d computational basis states $|\psi\rangle = \sum_{i=0}^{2^N-1} \alpha_i |i\rangle$, where $\sum_i |\alpha_i|^2 = 1$. The probability of observing the qubits in state $|i\rangle$ can be found by $|\alpha_i|^2$ using the Born rule [37]. Applying (noise-free) gates to the qubits corresponds to unitary time evolution of the state, i.e. $|\psi'\rangle = U|\psi\rangle$, where $U \in \mathbb{C}^{d\times d}$ is a unitary matrix. If U encapsulates the entire quantum circuit parameterized by the real parameters $\boldsymbol{\theta}$, then the input/output

relation can be written as $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})\,|\underline{0}\rangle$, where

$$U(\boldsymbol{\theta}) = U_P(\theta_P)U_{P-1}(\theta_{P-1})...U_1(\theta_1) \tag{5}$$

represent the gate sequence with $\boldsymbol{\theta} \in \mathbb{R}^P$ being a real vector containing all the circuit parameters. We denote $U_{\boldsymbol{\Theta}}(\boldsymbol{\theta})$ to be the quantum circuit unitary with hyperparameters $\boldsymbol{\Theta}$ and circuit parameters $\boldsymbol{\theta}$. By hyperparameters, we refer to the choice of circuit design (number of ansatz layers, choice of classical optimizer, learning rate, etc.).

At the end of the quantum circuit, the qubits are measured and the state collapses to one of the computational basis states. The measurement outcome is then used to estimate the expectation of a *parameterized* Hamiltonian $\mathcal{H}(\boldsymbol{C})$ as given by eq. (2) but for brevity we collect all Hamiltonian coefficients in $\boldsymbol{C} = \{\mathbf{J}, \mathbf{b}\}$. If the qubits are in state $|\psi(\boldsymbol{\theta})\rangle$, the expected value $\langle\mathcal{H}(\boldsymbol{C})\rangle$ can mathematically be found from "sandwiching" the matrix $\mathcal{H}(\boldsymbol{C})$ between the state and its complex conjugate transpose:

$$\langle\mathcal{H}(\boldsymbol{C})\rangle = \langle\psi(\boldsymbol{\theta})|\,\mathcal{H}(\boldsymbol{C})\,|\psi(\boldsymbol{\theta})\rangle \tag{6}$$

We note that $\mathcal{H} \in \mathbb{C}^{d\times d}$ is a Hermitian matrix and hence $\langle\mathcal{H}(\boldsymbol{C})\rangle \in \mathbb{R}$. If the task for the VQA is to minimize eq. (6) with respect to $\boldsymbol{\theta}$, we call it a *variational quantum eigensolver* (VQE) [7]. This corresponds to finding the parameter vector

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\,\langle\psi(\boldsymbol{\theta})|\,\mathcal{H}(\boldsymbol{C})\,|\psi(\boldsymbol{\theta})\rangle \tag{7}$$

using a classical optimizer, often gradient descent[ref]. More sophisticated methods such as imaginary-time evolution [38] — which for pure quantum states in unitary evolution is equivalent to natural gradient descent [39] — and other optimizers such as Adam[40] have also been used with great success. A very recent approach went beyond gradient-based methods by utilizing root-finding for operator covariances [41] which combined with classical shadows [42] opens up fundamentally new avenues for VQA optimization and research.

B. Bayesian Optimization

Bayesian Optimization (BO) [16] is well-suited for black-box optimization problem such as given a problem Hamiltonian with coefficients $\boldsymbol{C}$, find the best VQA hyperparameters $\boldsymbol{\Theta}^* \in \mathcal{S}$ in search region $\mathcal{S}$, which minimizes some loss function $\mathcal{L}$ conditioned on the problem Hamiltonian. Hence

$$\boldsymbol{\Theta}^* = \underset{\boldsymbol{\Theta}\in\mathcal{S}}{\operatorname{argmin}}\,[\mathcal{L}(\boldsymbol{\Theta}|\boldsymbol{C}) + \epsilon] \tag{8}$$

where ϵ is noise. Since our simulations are only concerned with VQE, we have $\mathcal{L}(\boldsymbol{\Theta}|\boldsymbol{C}) = \langle 0|U^\dagger_{\boldsymbol{\Theta}}(\boldsymbol{\theta})\mathcal{H}(\boldsymbol{C})U_{\boldsymbol{\Theta}}(\boldsymbol{\theta})|0\rangle$ Note here that some dimensions of $\boldsymbol{\Theta}$ might be discrete (e.g. number of ansatz layers) or categorical (e.g. choice of classical optimizer). Evaluating $\mathcal{L}$ might be expensive in the sense of e.g. monetary or time costs. We can therefore only obtain a finite collection of such input-output relations $\mathcal{D} = \{\boldsymbol{\Theta}_j, \langle\mathcal{H}\rangle_j\}_{j=1}^B$, where B is correspondingly small. The goal of BO is that a good set of hyperparameters for a particular problem Hamiltonian has been chosen after B iterations, and that this choice is better than the best one in a randomly selected pool of B hyperparameter configurations. BO requires 1) a choice of surrogate model, which is a machine learning model that learns a map $f : \boldsymbol{\Theta} \to \mathcal{L}(\boldsymbol{\Theta}|\boldsymbol{C})$, 2) acquisition function, which is the strategy of how to query new datapoints based on previously observed datapoints, and 3) a search-space and overall time/iteration budget. As we will elaborate on momentarily, we expand this model such that the $\mathcal{D}$ contains both VQA hyperparameters *and* the coefficients in the Hamiltonian, i.e. $\mathcal{D} = \{\boldsymbol{\Theta}_j, \boldsymbol{C}, \langle\mathcal{H}(\boldsymbol{C})\rangle_j\}_{j=1}^B$.

a. Surrogate model A popular choice of surrogate model is the Gaussian Process (GP) [43] as that often provides a good fit for relatively few datapoints by having the option of incorporating prior knowledge into the model via choice of kernel. In this work, we use the radial basis function (RBF) kernel and learn the kernel parameters using the log likelihood function of the GP. Although the input space is hybrid (a combination of categorical and continuous inputs), using the RBF kernel has numerically been shown to perform competitively against more advanced and computationally heavy kernels on real-world problems [44].

We refer to Ref. [43] for a thorough explanation of GPs, but mention that an important element is that they not only provide a prediction for new unseen y's, but also an uncertainty associated with this estimate. For BO this uncertainty is often used to query new datapoints via the acquisition function.

b. Acquisition function The acquisition function decides which new point x_i to query, using the prediction and associated uncertainty from the surrogate model. In our case, input x_i is the VQA hyperparameters. We use the expected improvement (EI) as defined by

$$\begin{aligned}&\mathrm{Acq}(\boldsymbol{\Theta}, \boldsymbol{C}|p(\langle\mathcal{H}\rangle\,|\boldsymbol{C}, \boldsymbol{\Theta}, \mathcal{D})) = \\ &(\boldsymbol{\mu}(\boldsymbol{\Theta}, \boldsymbol{C}) - f^*(\boldsymbol{\Theta}, \boldsymbol{C}))\Phi\Big(\frac{\boldsymbol{\mu}(\boldsymbol{\Theta}, \boldsymbol{C}) - f^*(\boldsymbol{\Theta}, \boldsymbol{C})}{\boldsymbol{\Sigma}(\boldsymbol{\Theta}, \boldsymbol{C})}\Big) \\ &+ \boldsymbol{\Sigma}(\boldsymbol{\Theta}, \boldsymbol{C})\phi\Big(\frac{\boldsymbol{\mu}(\boldsymbol{\Theta}, \boldsymbol{C}) - f^*(\boldsymbol{\Theta}, \boldsymbol{C})}{\boldsymbol{\Sigma}(\boldsymbol{\Theta}, \boldsymbol{C})}\Big)\end{aligned} \tag{9}$$

where Φ and ϕ are the cumulative and density function of the standard (multivariate) normal, the term $f^*(\boldsymbol{\Theta}, \boldsymbol{C})$ is smallest energy obtained so far for the Hamiltonian $\mathcal{H}(\boldsymbol{C})$, $p(\langle\mathcal{H}\rangle\,|\boldsymbol{C}, \boldsymbol{\Theta})$ is the surrogate posterior distribution trained on dataset $\mathcal{D}$ providing $\mu(\boldsymbol{\Theta}, \boldsymbol{C})$ as mean (prediction of $\langle\mathcal{H}\rangle$) and $\boldsymbol{\Sigma}(\boldsymbol{\Theta}, \boldsymbol{C})$ as covariance matrix (diagonal contains uncertainty estimate for corresponding prediction). The BO routine picks the hyperparameters according the the maximum of this function, that is, $\boldsymbol{\Theta}_{t+1} = \operatorname{argmax}_{\boldsymbol{\Theta}} \mathrm{Acq}(\boldsymbol{\Theta}, \boldsymbol{C})$. In the Appendix we elaborate on how to incorporate prior weight to specific

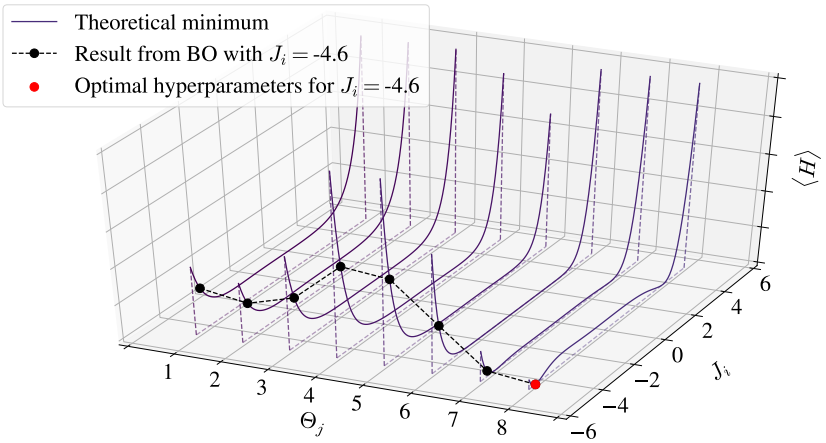


Figure 2: Illustration of the concept on fictitious problem. For a slice of a Hamiltonian, i.e. a specific problem with $J_i = -4.6$, we can try out different hyperparameters $\boldsymbol{\Theta}_j$ to minimize $\langle H \rangle$ using BO. When the procedure is done, we subsequently use AL to try a new J_i drawing knowledge from previous experiments to warm start the set of hyperparameters.

hyperparameter values.

C. Active Learning

Active Learning is almost equivalent to BO, except that the goal is not to find global min/max of a black-box function, but instead to "learn as much about the function" in as few steps as possible [45]. In a sense, we let the model select which data to learn from. The goal of for AL is similar to regression (manifold fitting), but "active" by means of having the possibility to iteratively query specific datapoints. In this work, we use a GP as surrogate, and share the information obtained in the BO routine. We use Uncertainty Sampling (UC) as querying strategy. Since we want to base the choice of next Hamiltonian with coefficients $\boldsymbol{C}_{t+1}$ on the uncertainty in the energy landscape as a function of $\boldsymbol{C}$, that is, $p(\langle \mathcal{H} \rangle | \boldsymbol{C})$, we can use Monte Carlo sampling to approximate this distribution from the surrogate output distribution of the agent GP, hence

$$\hat{p}(\langle \mathcal{H} \rangle | \boldsymbol{C}) = \sum_{k=1}^{K} p(\langle \mathcal{H} \rangle | \boldsymbol{C}, \boldsymbol{\Theta}_k) p(\boldsymbol{\Theta}_k) \tag{10}$$

where if $\boldsymbol{\Theta}_k \sim p(\boldsymbol{\Theta}_k)$ then $p(\boldsymbol{\Theta}_k) = \frac{1}{K}$. We use this distribution in the definition of uncertainty sampling

$$\boldsymbol{C}_{t+1} = \underset{\boldsymbol{C}}{\operatorname{argmax}} - \sum_i \hat{p}(\langle \mathcal{H} \rangle_i | \boldsymbol{C}) \log \hat{p}(\langle \mathcal{H} \rangle_i | \boldsymbol{C}) \tag{11}$$

For GPs this corresponds exactly to querying the point(s) for which the uncertainty estimate, i.e. the diagonal element(s) in the covariance matrix, is largest. Instead of picking the largest acquisition value, another strategy could be to sample $\boldsymbol{C}_{t+1}$ from a probability distribution with probability mass inversely proportional to the agent uncertainty. Hence by sampling points from this distribution, one can execute multiple experiments in parallel if suitable for the particular problem, which might accelerate the procedure but likely also require more experiments.

III. ACTIVE META-VQE

Our goal is to be able to hand the algorithmic agent a Hamiltonian with coefficients $\boldsymbol{C}$, and in return the agent outputs a set hyperparameters $\boldsymbol{\Theta}$ which is 1) better than a randomly sampled $\boldsymbol{\Theta}$ from the search space, 2) at least as good as the overall best hyperparameter settings and 3) an initial guess to circuit parameters $\boldsymbol{\theta}$. Here we quantify "better" with two metrics: infidelity (see eq. (12)) and regret (see eq. (13)). Our strategy for the agent is as follows. First query n_{init} initial points in the $\boldsymbol{C}$ subspace and run the VQA for each to get the agent "warmed up"; this collects the initial dataset $\mathcal{D} = \{\boldsymbol{C}_i, \{\{\boldsymbol{\Theta}_{ij}\}_{j=1}^{B}\}_i, \{\{\langle \mathcal{H} \rangle_{ij}\}_{j=1}^{B}\}_i\}_{i=1}^{n_{\text{init}}}$ via n_{init} independent BO routines. These BO routines can be done in parallel. For each of these BO routines i, we perform BO to get a <u>collection</u> (indexed with j) of hyperparameters and corresponding energy expectations, namely $\{\boldsymbol{\Theta}_{ij}\}$ and $\{\langle \mathcal{H} \rangle_{ij}\}$, obtained via gradient descent of the circuit parameters. Hence for every value of $\boldsymbol{C}_i$, which we can think of as a slice in the search space, we try out a bunch of different hyperparameters (via BO) and record the corresponding energy expectation. Next step is to begin the main loop of ALBO, which consists of iteratively alternating between Al and BO up to a maximum number of iterations T. Via our AL strategy, the agent picks the

next problem Hamiltonian at iteration $t = n_{\text{init}} + 1$, that is, $\mathcal{H}(\boldsymbol{C}_t)$ to investigate. Subsequently, the agent performs BO in order to find the best hyperparameters for *that* specific $\mathcal{H}(\boldsymbol{C}_t)$. The protocol is summarized in algorithm 1. Again, the crucial aspect here is that the agent gets access to more and more slices $\{\boldsymbol{C}_1, \boldsymbol{C}_2, ...\}$ and for each slice $\boldsymbol{C}_i$ has a collection of $\{\boldsymbol{\Theta}_j, \langle\mathcal{H}\rangle_j\}_{j=1}^{B}$ where B is the number of BO iterations. The goal for the agent is learn and use lessons across, between and to some extend away from slices and thus be able to come up with better VQA strategies for unseen problems. A pictorial scheme is provided in fig. 2 to illustrate the concept. Here 8 different hyperparameter configurations were applied to a specific Hamiltonian. The basic concept behind ALBO is that the minimum found by BO will (hopefully) be a better starting point when changing J_i, i.e. changing the Hamiltonian, than randomly picking $\boldsymbol{\Theta}_j$. The algorithm is summarized in algorithm 1.

In order to test these relatively broad yet potentially powerful conjectures consisting of relatively weak assumptions, we simulate the algorithm in algorithm 1 called ALBO as described in section IV A.

Algorithm 1 Actively Learned Bayesianly Optimized (ALBO) VQE.

Require: Search space $\mathcal{S}$, iterations $\{n_{\text{AL}}, n_{\text{BO}}, n_{\text{VQE}}\}$, prior $p(\boldsymbol{\Theta})$
  Initialize AGENT with $\mathcal{S}$
  Get n_{init} samples: $\{\boldsymbol{C}_i\}_{i=1}^{n_{\text{init}}} \leftarrow \text{Uniform}(\mathcal{S}_{\boldsymbol{C}})$
  Obtain $\mathcal{D} = \{\boldsymbol{C}_i, \{\{\boldsymbol{\Theta}_j\}_{j=1}^{n_{\text{BO}}}\}_i, \{\{\langle\mathcal{H}(\boldsymbol{C}_i)\rangle_j\}_{j=1}^{n_{\text{BO}}}\}_i\}_{i=1}^{n_{\text{init}}}$
  Update AGENT surrogate with $\mathcal{D}$
Procedure:
  for $t \leq n_{\text{AL}}$ **do** ▷ AL
    Compute $\hat{p}(\langle\mathcal{H}\rangle|\boldsymbol{C})$ via eq. (10)
    Get $\boldsymbol{C}_t \leftarrow \text{AGENT}(\hat{p}(\langle\mathcal{H}\rangle|\boldsymbol{C}))$ via eq. (11)
    Set $\mathcal{H} := \mathcal{H}(\boldsymbol{C}_t)$
    for $j \leq n_{\text{BO}}$ **do** ▷ BO
      Get $\boldsymbol{\Theta}_j \leftarrow \text{AGENT}(p(\langle\mathcal{H}\rangle|\boldsymbol{C}, \boldsymbol{\Theta}))$ eq. (9)
      Get $\boldsymbol{\theta}_0 \leftarrow \text{argmin}_{\boldsymbol{\theta}_k} K(\mathcal{D}_k, \mathcal{D})$
      Initialize VQA $U_{\boldsymbol{\Theta}}(\boldsymbol{\theta}_0)$
      Set $\mathcal{L}(\boldsymbol{\Theta}_j|\boldsymbol{C}_t) := \langle 0|U^\dagger_{\boldsymbol{\Theta}_j}(\boldsymbol{\theta})HU_{\boldsymbol{\Theta}_j}(\boldsymbol{\theta})|0\rangle$
      for $i \leq n_{\text{VQE}}$ **do** ▷ VQE
        Get $\langle\mathcal{H}\rangle_i, \boldsymbol{\theta}_i \leftarrow \text{VQA}(\mathcal{L}(\boldsymbol{\Theta}_j|\boldsymbol{C}_t), \boldsymbol{\theta}_{i-1})$
        Compute $\mathcal{I}$ via eq. (12)
      Compute $\mathcal{R}$ via eq. (13)
      Set $\langle H\rangle_j := \langle\mathcal{H}\rangle_{n_{\text{VQE}}}$
      Update $\mathcal{D}$ with $\{\boldsymbol{C}_t, \boldsymbol{\Theta}_j, \langle\mathcal{H}\rangle_j\}$
      Update AGENT surrogate with $\mathcal{D}$

IV. EXPERIMENTAL RESULTS

In this section, we first outline the simulation strategy in section IV A followed by experiments transverse-field Ising model in section IV B and the Heisenberg XXZ model in section IV C.

A. Simulation Strategy

We test ALBO up against a variety of alternatives: random search (**RS+RS**) and random search + Bayesian optimization (**RS+BO**), where (**X+Y**) refers to the strategy **X** in the $\mathcal{S}_{\boldsymbol{C}}$ subspace plus the strategy **Y** in the $\mathcal{S}_{\boldsymbol{\Theta}}$ subspace, respectively. In the following results, we report the infidelity with the true groundstate $|\psi_0\rangle$ as given by

$$\mathcal{I} = 1 - |\langle\psi_0|\psi(\boldsymbol{\theta})\rangle|^2 \tag{12}$$

where $\psi(\boldsymbol{\theta})$ is the state at the end of the circuit optimization procedure. We also report the total Regret $\mathcal{R}$ defined by

$$\mathcal{R} = \sum_{i=1}^{n_{\text{iter}}} 1 - \left|\left\langle\psi_0\middle|\psi(\boldsymbol{\theta}^{(i)})\right\rangle\right|^2 \tag{13}$$

where n_{iter} is the number of variational circuit optimization iterations. The infidelity reveals how good the final state is, whereas the regret reveals is a measure of how good the groundstate is found combined with how fast the circuit parameters converge. As a proof of concept, we simulate three qubit systems, and investigate the Ising chain (IC), transverse field Ising chain (TFI) and Heisenberg (HB) spin models together with the Gradient Descent Optimizer. We use L (part of hyperparameter tuning) layers of the ansatz. We run $T = 10$ active learning iterations, $B = 10$ Bayesian optimization iterations and $K = 100$ circuit parameter updates. The search regions for the Hamiltonian coefficients and hyperparameters are, respectively,

$$\begin{aligned}
&\mathcal{S}_{\boldsymbol{C}} = \{C : C \in \mathbb{R}^{|\boldsymbol{C}|}, -1 \leq [C]_i \leq 1 \ \forall \ i\} \\
&\mathcal{S}_{\boldsymbol{\Theta}} = \{ \\
&\quad \{\eta : \eta \in \mathbb{R}, 10^{-5} \leq \eta \leq 10^{-1}\}, \\
&\quad \{L : L \in \mathbb{Z}, 1 \leq L \leq 10\}, \\
&\quad \}
\end{aligned}$$

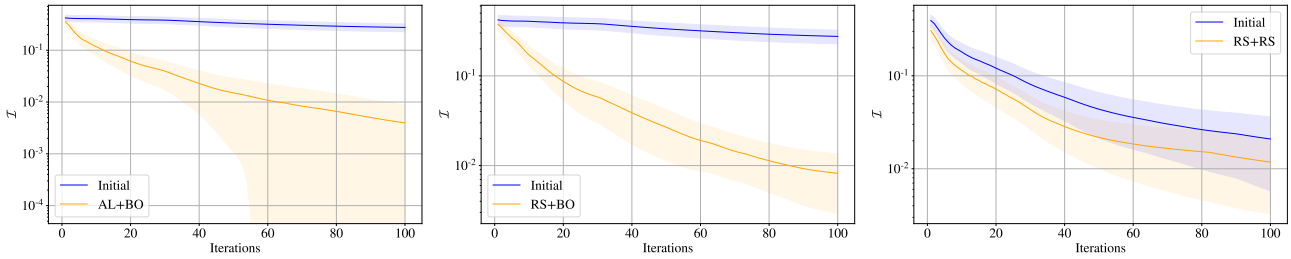
where η is the learning rate, L is the number of ansatz layers. All simulations are done with Pennylane [46] and with GP implementations in scikit-learn [47]. Code is available on github: `https://github.com/jfold/albo`.

B. Transverse-field Ising Chain

The Hamiltonian for the Transverse-field Ising (TFI) Chain is given by

$$\mathcal{H}_{TFI} = -\sum_i b_i Z_i - \sum_{<i,j>} J_{i,j} Z_i Z_j - \sum_i h_i X_i, \tag{14}$$

where Z_i and X_i are the Pauli Z and X operators, respectively, acting on the i'th qubit. In fig. 3, we show the result of running ALBO (a), RS+BO (B) and RS+RS (c)



(a) Active learning + Bayesian optimization (AL+BO)

(b) Random search + Bayesian optimization (RS+BO)

(c) Random search + Random search (RS+RS)

Figure 3: TFI Simulation of agent for various strategies. We observe ALBO beats both RS+BO and all methods improve after initialization (blue).

on TFI test models. We see that across problems, the agent when using both AL and BO outperforms random search in the Hamiltonian space combined with BO in the hyperparameter space.

C. Spin Hamiltonian: 1D XXZ model

The 1D antiferromagnetic XXZ spin Hamiltonian is given by

$$\mathcal{H} = \sum_{i=1}^{n} \alpha Z_i Z_{i+1} + Y_i Y_{i+1} + X_i X_{i+1} + \lambda \sum_{i=1}^{n} Z_i \quad (15)$$

with two parameters α and λ, and where $\{X_i, Y_i, Z_i\}$ refers to the Pauli-X,Y,Z observables acting on qubit i, respectively. fig. 4 show test results for the XXZ and again we see that AL+BO outperforms random search even with BO in the hyperparameter space. Furthermore, we also compare with an often used standard hyperparameter (SH) heuristic, that of $\lceil \frac{N}{2} \rceil$ layers and learning rate $\eta = 0.4$. In fig. 5a, results are displayed and we see that AL+BO achieves significantly better regret than all of the other strategy combinations.

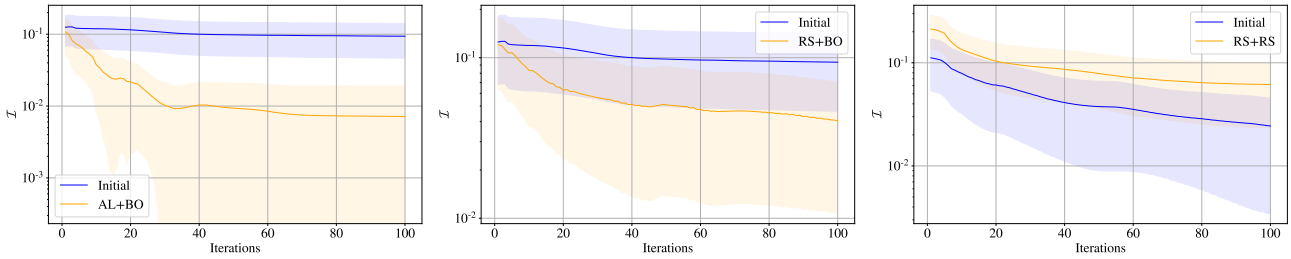
V. CONCLUSION

In this paper, we proposed a new classical algorithm that can guide experimental design across various problems / Hamiltonians. In all experiments, the meta-learner outperformed random search and standard heuristics which suggest evidence of learnable patterns in the space of hyperparameters and hamiltonian coefficients. Even with a vanilla Gaussian Process which assumes normally distributed objective function works well. An interesting question to be analyzed in further work is how far apart the loss functions can be and how few initial datapoints we can get away with in order to benefit from the classical agent, that is, the number of initial points in active learning part seemed crucial to get a good performance. This is not an unknown phenomenon as there is a "Goldilocks Zone" for when active learning yields good results. Other potential ideas could include penalizing agent's search space by putting prior weight on shallower circuits. In the simulations, we used the RBF kernel, but more advanced kernels might be even more suitable as we are dealing with a hybrid and partly categorical input space. A big leap forward would be if we could use parameters to warm start test problems, however, this is not trivial for several reasons. First, close in state space is not the same as close in parameter space. Second, the hyperparameters suggested by the classical agent might not be one that is in the training set, that is, we do not posses the circuit parameters which prepares the solution. One could think of adiabatic-like ideas where, if we had the circuit parameter solution to one Hamiltonian, one could slowly morph these into the solution for another unseen Hamiltonian. Instead of taking the maximum of acquisitions and having access to both $p(\boldsymbol{\Theta})p(\boldsymbol{\Theta}|\boldsymbol{C})$, we can sample from this distribution instead. This is known as posterior sampling, and might be an interesting future direction — Appendix A suggest how one could do so.

Future work should investigate larger systems and how one can apply ALBO to a specific search region of interest. For example are there scenarios where the coefficients are not known exactly but their distribution $p(\mathbf{J}, \mathbf{b})$ is known. Finding the optimal hyperparameters on samples from this distribution will lead to learning which hyperparameters work well on for that distribution. We thus hope that future work would exploit classical algorithmic agents like this in order to advance interesting quantum computing research. Our framework is very general and in principle allow for any loss function and search regions.

ACKNOWLEDGEMENT

The author thanks Lars Kai Hansen and Bálint Koczor for discussions and feedback on manuscript. JF author was supported by William Demant Foundation PhD Scholarship [Grant Number 18-4438].

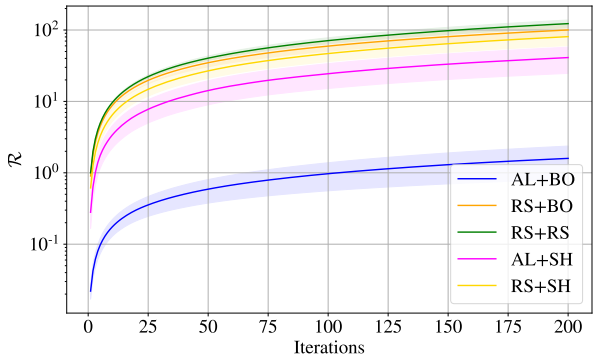


(a) Active learning + Bayesian optimization (AL+BO)

(b) Random search + Bayesian optimization (RS+BO)

(c) Random search + Random search (RS+RS)

Figure 4: XXZ Simulation of agent for various strategies. We observe ALBO beats both RS+BO and all methods improve after initialization (blue), except for the RS+RS strategy.



(a)

Figure 5: XXZ Regret as a function of iterations averaged over seeds. We observe ALBO beating all other strategy combinations.

[1] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

[2] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, 2015.

[3] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[4] Scott Aaronson. Read the fine print. *Nature Physics*, 11(4):291–293, 2015.

[5] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S. Kottmann, Tim Menke, Wai-Keong Mok, Sukin Sim, Leong-Chuan Kwek, and Alán Aspuru-Guzik. Noisy intermediate-scale quantum (nisq) algorithms, 2021.

[6] Alejandro Perdomo-Ortiz, Marcello Benedetti, John Realpe-Gómez, and Rupak Biswas. Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers. *Quantum Science and Technology*, 3(3):030502, 2018.

[7] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5(1):1–7, 2014.

[8] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[9] Jonathan Foldager, Arthur Pesah, and Lars Kai Hansen. Noise-assisted variational quantum thermalization. *Scientific reports*, 12(1):3862, 2022.

[10] Guillaume Verdon, Jacob Marks, Sasha Nanda, Stefan Leichenauer, and Jack Hidary. Quantum hamiltonian-based models and the variational quantum thermalizer algorithm. *arXiv preprint arXiv:1910.02071*, 2019.

[11] Michael Lubasch, Jaewoo Joo, Pierre Moinier, Martin Kiffner, and Dieter Jaksch. Variational quantum algorithms for nonlinear problems. *Physical Review A*, 101(1):010301, 2020.

[12] Xiaosi Xu, Jinzhao Sun, Suguru Endo, Ying Li, Simon C Benjamin, and Xiao Yuan. Variational algorithms for linear algebra. *Science Bulletin*, 66(21):2181–2188, 2021.

[13] Tao Xin, Liangyu Che, Cheng Xi, Amandeep Singh, Xinfang Nie, Jun Li, Ying Dong, and Dawei Lu. Experimental quantum principal component analysis via parametrized quantum circuits. *Physical Review Letters*, 126(11):110502, 2021.

[14] Vicente Leyton-Ortega, Alejandro Perdomo-Ortiz, and Oscar Perdomo. Robust implementation of generative modeling with parametrized quantum circuits. *arXiv preprint arXiv:1901.08047*, 2019.

[15] Sukin Sim, Peter D Johnson, and Alán Aspuru-Guzik. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, 2(12):1900070, 2019.

[16] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *arXiv preprint arXiv:1206.2944*, 2012.

[17] Alba Cervera-Lierta, Jakob S Kottmann, and Alán Aspuru-Guzik. Meta-variational quantum eigensolver: Learning energy profiles of parameterized hamiltonians for quantum simulation. *PRX Quantum*, 2(2):020329, 2021.

[18] Eric R Anschuetz and Bobak T Kiani. Quantum variational algorithms are swamped with traps. *Nature Communications*, 13(1):7760, 2022.

[19] Arthur Pesah, M Cerezo, Samson Wang, Tyler Volkoff, Andrew T Sornborger, and Patrick J Coles. Absence of barren plateaus in quantum convolutional neural networks. *arXiv preprint arXiv:2011.02966*, 2020.

[20] Louis Schatzki, Martin Larocca, Frederic Sauvage, and Marco Cerezo. Theoretical guarantees for permutation-equivariant quantum neural networks. *arXiv preprint arXiv:2210.09974*, 2022.

[21] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature communications*, 9(1):1–6, 2018.

[22] Tobias Haug, Kishor Bharti, and M. S. Kim. Capacity and quantum geometry of parametrized quantum circuits, 2021.

[23] Vishwanathan Akshay, Daniil Rabinovich, Ernesto Campos, and Jacob Biamonte. Parameter concentrations in quantum approximate optimization. *Physical Review A*, 104(1):L010401, 2021.

[24] Harper R Grimsley, Sophia E Economou, Edwin Barnes, and Nicholas J Mayhall. An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nature communications*, 10(1):1–9, 2019.

[25] Ho Lun Tang, Edwin Barnes, Harper R Grimsley, Nicholas J Mayhall, and Sophia E Economou. qubit-adapt-vqe: An adaptive algorithm for constructing hardware-efficient ansatze on a quantum processor. *arXiv preprint arXiv:1911.10205*, 2019.

[26] Mateusz Ostaszewski, Edward Grant, and Marcello Benedetti. Quantum circuit structure learning. *arXiv preprint arXiv:1905.09692*, 2019.

[27] Arthur G Rattew, Shaohan Hu, Marco Pistoia, Richard Chen, and Steve Wood. A domain-agnostic, noise-resistant, hardware-efficient evolutionary variational quantum eigensolver. *arXiv*, pages arXiv–1910, 2019.

[28] Tim Menke, Florian Häse, Simon Gustavsson, Andrew J Kerman, William D Oliver, and Alán Aspuru-Guzik. Automated design of superconducting circuits and its application to 4-local couplers. *npj Quantum Information*, 7(1):1–8, 2021.

[29] Guillaume Verdon, Michael Broughton, Jarrod R McClean, Kevin J Sung, Ryan Babbush, Zhang Jiang, Hartmut Neven, and Masoud Mohseni. Learning to learn with quantum neural networks via classical neural networks. *arXiv preprint arXiv:1907.05415*, 2019.

[30] Max Wilson, Rachel Stromswold, Filip Wudarski, Stuart Hadfield, Norm M Tubman, and Eleanor G Rieffel. Optimizing quantum heuristics with meta-learning. *Quantum Machine Intelligence*, 3(1):1–14, 2021.

[31] Shi-Xin Zhang, Chang-Yu Hsieh, Shengyu Zhang, and Hong Yao. Differentiable quantum architecture search. *arXiv preprint arXiv:2010.08561*, 2020.

[32] Mohammad Pirhooshyaran and Tamas Terlaky. Quantum circuit design search. *arXiv preprint arXiv:2012.04046*, 2020.

[33] Ali Rad, Alireza Seif, and Norbert M Linke. Surviving the barren plateau in variational quantum circuits with bayesian learning initialization. *arXiv preprint arXiv:2203.02464*, 2022.

[34] Kevin J Sung, Jiahao Yao, Matthew P Harrigan, Nicholas C Rubin, Zhang Jiang, Lin Lin, Ryan Babbush, and Jarrod R McClean. Using models to improve optimizers for variational quantum algorithms. *Quantum Science and Technology*, 5(4):044008, 2020.

[35] Giovanni Iannelli and Karl Jansen. Noisy bayesian optimization for variational quantum eigensolvers. *arXiv preprint arXiv:2112.00426*, 2021.

[36] Alexey A Melnikov, Hendrik Poulsen Nautrup, Mario Krenn, Vedran Dunjko, Markus Tiersch, Anton Zeilinger, and Hans J Briegel. Active learning machine learns to create new quantum experiments. *Proceedings of the National Academy of Sciences*, 115(6):1221–1226, 2018.

[37] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

[38] Sam McArdle, Tyson Jones, Suguru Endo, Ying Li, Simon C Benjamin, and Xiao Yuan. Variational ansatz-based quantum simulation of imaginary time evolution. *npj Quantum Information*, 5(1):1–6, 2019.

[39] James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. Quantum natural gradient. *Quantum*, 4:269, 2020.

[40] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[41] Gregory Boyd and Bálint Koczor. Training variational quantum circuits with covar: covariance root finding with classical shadows. *arXiv preprint arXiv:2204.08494*, 2022.

[42] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, 2020.

[43] Christopher K Williams and Carl Edward Rasmussen. Gaussian processes for machine learning, vol. 2. *MA: MIT press Cambridge*, 2006.

[44] Aryan Deshwal, Syrine Belakaria, and Janardhan Rao Doppa. Bayesian optimization over hybrid spaces. In *International Conference on Machine Learning*, pages 2632–2643. PMLR, 2021.

[45] Burr Settles. Active learning literature survey. 2009.

[46] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, Carsten Blank, Keri McKiernan, and Nathan Killoran. Pennylane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968*, 2018.

[47] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vin-

cent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Appendix A: Incorporating priors to hyperparameter queries

Recall that the expected improvement acquisition function is defined by

$$\begin{aligned}\mathrm{Acq}(\boldsymbol{\Theta}, \boldsymbol{C}) =& (\boldsymbol{\mu}(\boldsymbol{\Theta}, \boldsymbol{C}) - f^*(\boldsymbol{\Theta}, \boldsymbol{C}))\Phi\Big(\frac{\boldsymbol{\mu}(\boldsymbol{\Theta}, \boldsymbol{C}) - f^*(\boldsymbol{\Theta}, \boldsymbol{C})}{\boldsymbol{\Sigma}(\boldsymbol{\Theta}, \boldsymbol{C})}\Big) \\ &+ \boldsymbol{\Sigma}(\boldsymbol{\Theta}, \boldsymbol{C})\phi\Big(\frac{\boldsymbol{\mu}(\boldsymbol{\Theta}, \boldsymbol{C}) - f^*(\boldsymbol{\Theta}, \boldsymbol{C})}{\boldsymbol{\Sigma}(\boldsymbol{\Theta}, \boldsymbol{C})}\Big)\end{aligned} \tag{S1}$$

Instead of taking the maximum argument $\boldsymbol{\Theta}$ of this quantity, we can normalize $\mathrm{EI}(\boldsymbol{\Theta})$ and thereby interpret this as a probability distribution

$$p(\boldsymbol{\Theta}|\boldsymbol{C}) = \frac{\mathrm{Acq}(\boldsymbol{\Theta}, \boldsymbol{C})}{\int_{\boldsymbol{\Theta}} \mathrm{Acq}(\boldsymbol{\Theta}, \boldsymbol{C})} \tag{S2}$$

In practice, we do so with a finite collection of $\boldsymbol{\Theta} = [\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, ...\boldsymbol{\Theta}_K]$ such that the numerator becomes a sum, that is,

$$p(\boldsymbol{\Theta}_j|\boldsymbol{C}) = \frac{\mathrm{Acq}(\boldsymbol{\Theta}_j, \boldsymbol{C})}{\sum_{k=1}^{K} \mathrm{Acq}(\boldsymbol{\Theta}_k, \boldsymbol{C})}. \tag{S3}$$

Given a prior $p(\boldsymbol{\Theta})$ distribution, we can find the maximum argument of the product of these two probability distributions:

$$\boldsymbol{\Theta}_{\mathrm{MAP}} = \underset{\boldsymbol{\Theta}}{\mathrm{argmax}}\, p(\boldsymbol{\Theta})p(\boldsymbol{\Theta}|\boldsymbol{C}). \tag{S4}$$

and thereby obtain a "MAP-like" Bayesian decision.

Appendix D

Paper D (under review)

THE FOLLOWING PAGES CONTAIN A COPY OF THE PREPRINT VERSION OF THE PAPER CURRENTLY UNDER REVIEW

ON THE ROLE OF MODEL UNCERTAINTIES IN BAYESIAN OPTIMIZATION

Jonathan Foldager
Section for Cognitive Systems, DTU Compute
Technical University of Denmark
Shared first authorship

Mikkel Jordahn
Section for Cognitive Systems, DTU Compute
Technical University of Denmark
Shared first authorship

Lars Kai Hansen
Section for Cognitive Systems, DTU Compute
Technical University of Denmark

Michael Riis Andersen
Section for Cognitive Systems, DTU Compute
Technical University of Denmark

April 11, 2023

ABSTRACT

Bayesian Optimization (BO) is a popular method for black-box optimization, which relies on uncertainty as part of its decision-making process when deciding which experiment to perform next. However, not much work has addressed the effect of uncertainty on the performance of the BO algorithm and to what extent calibrated uncertainties improve the ability to find the global optimum. In this work, we provide an extensive study of the relationship between the BO performance (regret) and uncertainty calibration for popular surrogate models and acquisition functions, and compare them across both synthetic and real-world experiments. Our results show that Gaussian Processes, and more surprisingly, Deep Ensembles are strong surrogate models. Our results further show a positive association between calibration error and regret, but interestingly, this association disappears when we control for the type of surrogate model in the analysis. We also study the effect of recalibration and demonstrate that it generally does not lead to improved regret. Finally, we provide theoretical justification for why uncertainty calibration might be difficult to combine with BO due to the small sample sizes commonly used.

1 Introduction

Probabilistic machine learning provides a framework in which it is possible to reason about uncertainty for both models and predictions [Ghahramani, 2015]. It is often argued that especially in high-stakes applications (healthcare, robotics, etc.), uncertainty estimates for decisions/predictions should be a central component and that they should be well-calibrated [Kuleshov and Deshpande, 2022]. The intuition behind calibration is that the uncertainty estimates should accurately reflect reality; for example, if a classification model predicts an 80% probability of belonging to class A on 10 datapoints, then (on average) we would expect 8 of those 10 samples actually belong to class A. Likewise – but less intuitively – in regression, if a calibrated model generates a prediction μ and uncertainty estimate σ, we would see p percent of the data lying inside a p percentile confidence interval of μ [Busk et al., 2021].

Uncertainty also plays a central role in Bayesian Optimization (BO) [Snoek et al., 2012], which will be the focus of this paper. As a sequential design strategy for global optimization, BO has several applications with perhaps the most popular ones being general experimental design [Shahriari et al., 2015] and model selection for machine learning models [Bergstra et al., 2011]. BO is most often used when the objective function is expensive (e.g. monetary, time-consuming, or ethically) to evaluate, gradients between in- and outputs are not available, noisy, and/or data acquisition is limited to few training samples [Agnihotri and Batra, 2020]. A BO protocol works by iteratively fitting a probabilistic surrogate model to observed values of an objective function, and using a so-called acquisition function (AF) based on the surrogate model, to select where to query the objective function next. In AFs, there is an inherent trade-off between exploring

input areas in which the surrogate model is uncertain of the underlying objective function, and exploiting areas where the surrogate model already knows that the objective value is low. As such, it seems obvious that in order for this exploration-exploitation trade-off to be good, the probabilistic model must be well-calibrated. It is, however, still not well-described how much calibration actually affects BO procedures. One could imagine that if calibration leads to a better model representation of the underlying objective function, as would be the general intuition, it would be natural to expect that improving calibration via so-called *recalibration* [Kuleshov et al., 2018] will aid in finding the global optimum of that same function.

1.1 Our Contribution

In this paper, we set out to investigate how the model uncertainties affect BO performance by means of both numerical and theoretical perspectives. Our work is highly motivated by the general intuition and understanding in the community that BO surrogate models with better / well-calibrated uncertainty estimates will perform better (i.e. reach better final and/or total regret). In particular, our paper is concerned with studying statements such as "BO crucially relying on calibrated uncertainty estimates" [Springenberg et al., 2016] and that methods performing worse "due to their frequentist uncertainty estimates" [Deshwal et al., 2021]. But how well-calibrated do we need to be in order to achieve good BO performance? In order to investigate these questions, we provide four major contributions:

- An extensive study of commonly used surrogate models and acquisition functions, where we study the resulting calibration errors and regrets to assess the relationship between calibration and regret. This includes an intervention study, where we manipulate model calibration and study the effect on regret.
- We show that Deep Ensembles is superior for hyperparameter tuning using BO.
- An investigation of whether recalibration during the BO protocol leads to better BO performance?
- Numerical and theoretical results to substantiate a discussion on the role of calibration in BO. Especially on the relationship between the number of recalibration samples and the variance of the calibration curve.

1.2 Related Work

A great deal of work has been carried out for uncertainty calibration for regression models [Kuleshov et al., 2018, Song et al., 2019, Ovadia et al., 2019, Busk et al., 2021, Nado et al., 2021] and the useful uncertainty toolbox [Chung et al., 2021] makes it easy to assess the calibration level of various models. In the very recent work by Deshpande and Kuleshov [2021], a procedure for calibrating Gaussian processes (GPs) during BO was proposed. Given the small sample sizes available in BO, the idea is to use leave-one-out cross-validation and utilize the calibration algorithm proposed in earlier work by Kuleshov et al. [2018]. We note that potential issues might arise from this procedure as the earlier work by [Kuleshov et al., 2018] states multiple times their approach produces calibrated forecasts "*given enough i.i.d. data*". However, the data available during BO is rarely large nor independent and identically distributed (i.i.d.), and the goal of our work is to dive deeper into this. Other research on the role of uncertainty calibration includes examples such as the work by Bliznyuk et al. [2008], where the authors propose a way of using Markov Chain Monte Carlo (MCMC) to get calibrated predictions for GPs. In Belakaria et al. [2020], the authors investigate uncertainty-aware multi-objective (multidimensional output) BO and argue that due to the uncertainty incorporating strategy, their model outperforms state-of-the-art procedures.

2 Background

Bayesian Optimization (BO) is concerned with the optimization task of finding the global minimum $\mathbf{x}^* = [x_1^*, x_2^*, ..., x_D^*]^\top$ of some objective function $f(\mathbf{x})$, where $\mathbf{x}$ is a D-dimensional vector, i.e.

$$\mathbf{x}^* = \operatorname{argmin} f(\mathbf{x}). \tag{1}$$

We assume that the optimization objective $f(\mathbf{x}) \in \mathbb{R}$ is contaminated with noise, i.e. we observe $y(\mathbf{x}) = f(\mathbf{x}) + \epsilon$, where ϵ is additive noise often assumed to follow an isotropic normal distribution. In many scenarios such as hyperparameter tuning of neural networks, the set of input variables $\mathbf{x}$ are rarely all continuous, and often no closed-form expression for f exists. Hence, BO is well-suited when f is a so-called "black-box" function [Turner et al., 2021]. At least two crucial decisions are to be made when using BO in practice: 1) the choice of surrogate model, which is to learn the underlying objective function f, and 2) the acquisition function (AF), which controls the strategy for deciding which input $\mathbf{x}$ to sequentially pick by maximizing the AF. Popular choices for surrogate models include Gaussian Processes (GPs) [Rasmussen, 2003, Snoek et al., 2012] and Random Forests (RFs) [Bergstra et al., 2011], but any model with a probabilistic interpretation, e.g. Deep Ensembles (DEs) [Lakshminarayanan et al., 2017] or mean-field Bayesian Neural Networks (BNNs) [Springenberg et al., 2016], can be used.

Table 1: BO results for experiments with synthetic data. For each of the surrogate and acquisition pairs here, we ran a total of 128 optimization problems, where each problem is repeated with 100 different seeds. For each pair, we report the mean of all $128 \cdot 100 = 12800$ runs and the standard error of the mean for all metrics. The instantaneous and total regret metrics are computed using eq. (8) and (9), respectively. ECE is the expected calibration error and is computed using eq. (7) and sharpness denotes the negatige entropy of the predictive distributions. Rows with Acquisition=Average (AVG) correspond to an average over all three acquisition strategies (EI, UCB, TS), but excluding random sampling (RS). Best performing configurations in each of the three sections (i.e. RS, EI+UCB+TS, AVG) are reported in bold font..

Surrogate	Acquisition	Inst. Regret	Total Regret	ECE	Sharpness
GP	RS	**0.496 ± 0.018**	**67.117 ± 2.155**	**0.005 ± 0.000**	-0.183 ± 0.012
DE	RS	0.508 ± 0.019	67.345 ± 2.194	0.011 ± 0.000	0.030 ± 0.007
RF	RS	0.511 ± 0.018	67.920 ± 2.205	0.006 ± 0.000	-0.478 ± 0.016
BNN	RS	0.519 ± 0.019	67.990 ± 2.199	0.088 ± 0.001	1.253 ± 0.008
GP	EI	0.036 ± 0.001	13.214 ± 0.325	0.016 ± 0.000	-0.224 ± 0.012
DE	EI	0.043 ± 0.002	21.714 ± 0.524	0.029 ± 0.001	-0.353 ± 0.009
RF	EI	0.099 ± 0.004	33.511 ± 0.994	0.025 ± 0.000	-0.386 ± 0.016
BNN	EI	0.848 ± 0.026	91.221 ± 2.719	0.113 ± 0.001	0.602 ± 0.008
GP	UCB	**0.027 ± 0.001**	**12.829 ± 0.328**	0.017 ± 0.000	-0.322 ± 0.012
DE	UCB	0.046 ± 0.002	21.148 ± 0.508	0.028 ± 0.001	-0.375 ± 0.009
RF	UCB	0.081 ± 0.003	31.173 ± 0.945	0.025 ± 0.000	-0.404 ± 0.016
BNN	UCB	0.480 ± 0.016	64.604 ± 1.830	0.097 ± 0.001	0.861 ± 0.007
GP	TS	0.041 ± 0.003	28.729 ± 1.044	**0.010 ± 0.000**	-0.436 ± 0.011
DE	TS	0.042 ± 0.002	22.116 ± 0.508	0.027 ± 0.001	-0.333 ± 0.009
RF	TS	0.279 ± 0.013	51.166 ± 1.783	0.013 ± 0.000	-0.451 ± 0.015
BNN	TS	0.628 ± 0.021	76.086 ± 2.330	0.091 ± 0.001	0.997 ± 0.007
GP	AVG	**0.035 ± 0.001**	**18.257 ± 0.390**	**0.015 ± 0.000**	-0.327 ± 0.007
DE	AVG	0.044 ± 0.001	21.659 ± 0.296	0.028 ± 0.000	-0.354 ± 0.005
RF	AVG	0.153 ± 0.005	38.616 ± 0.757	0.021 ± 0.000	-0.414 ± 0.009
BNN	AVG	0.652 ± 0.013	77.303 ± 1.346	0.100 ± 0.001	0.820 ± 0.005

Acquisition Functions For the choice of AF, *Expected Improvement* (EI) as proposed by Jones et al. [1998] is often used and is defined as follows:

$$\text{EI}(\mathbf{x}) = (\mu(\mathbf{x}) - f(\mathbf{x}^+))\Phi(Z) + \sigma(\mathbf{x})\phi(Z), \tag{2}$$

if $\sigma(\mathbf{x}) > 0$ otherwise $\text{EI}(\mathbf{x}) = 0$, and with $Z(\mathbf{x}) = \frac{\mu(\mathbf{x}) - f(\mathbf{x}^+)}{\sigma(\mathbf{x})}$, where $\mu(\mathbf{x})$ and $\sigma(\mathbf{x})$ denote the mean and standard deviation, respectively, of the surrogate function at $\mathbf{x}$, $f(\mathbf{x}^+)$ denotes the best function value observed so far, and Φ and ϕ denote the cumulative distribution function (CDF) and probability density function (PDF) of a standard normal distribution, respectively. Another popular AF is the *Upper Confidence Bound* (UCB), proposed in Srinivas et al. [2012] which is defined as:

$$\text{UCB}(\mathbf{x}) = -\mu(\mathbf{x}) + \beta^{1/2}\sigma(\mathbf{x}), \tag{3}$$

for minimization problems, where $\mu(\mathbf{x})$ and $\sigma(\mathbf{x})$ once again denote the mean and standard deviation of the surrogate function at $\mathbf{x}$ and β is a hyperparameter controlling the trade-off between exploitation and exploration. Finally, the acquisition strategy coined *Thompson Sampling* (Thompson [1933]) works by generating a random sample from the posterior of f and then locating the optimal value for the specific sample, i.e. for some sample $f(\mathbf{x}) \sim p(f|D)$

$$\text{TS}(\mathbf{x}) = -f(\mathbf{x}). \tag{4}$$

For GPs and BNNs this is done by sampling a function from the posterior, whilst for DEs and RFs we sample a neural network or tree, respectively (Elmachtoub et al. [2017]).

Calibration Following the work by Kuleshov et al. [2018], a regression model is well-calibrated if approximately q percent of the time test samples fall inside a q percent confidence interval of the predictive distribution. For regression

Table 2: BO results for hyperparameter tuning experiments. For each of the surrogate and acquisition pairs here, we ran a total of 6 optimization problems, where each problem is repeated with 100 different seeds. For each pair, we report the mean of all $6 \cdot 100 = 600$ runs and the standard error of the mean for all metrics. The instantaneous and total regret metrics are computed using eq. (8) and (9), respectively. ECE is the expected calibration error and is computed using eq. (7) and sharpness denotes the negative entropy of the predictive distributions. Rows with Acquisition=Average (AVG) correspond to an average over all three acquisition strategies (EI, UCB, TS), but excluding random sampling (RS). Best performing configurations in each of the three sections (i.e. RS, EI+UCB+TS, AVG) are reported in bold font.

Surrogate	Acquisition	Inst. Regret	Total Regret	ECE	Sharpness
GP	RS	0.0151 ± 0.0006	2.7021 ± 0.0995	$\mathbf{0.0055 \pm 0.0001}$	-0.7762 ± 0.0138
DE	RS	0.0161 ± 0.0007	2.7822 ± 0.1033	0.0093 ± 0.0001	-0.2574 ± 0.0134
RF	RS	0.0152 ± 0.0007	2.6977 ± 0.1018	0.0072 ± 0.0002	1.0302 ± 0.1017
BNN	RS	$\mathbf{0.0150 \pm 0.0007}$	$\mathbf{2.5948 \pm 0.0942}$	0.1015 ± 0.0005	1.3499 ± 0.0102
GP	EI	0.0031 ± 0.0002	1.5375 ± 0.0565	0.0153 ± 0.0004	-0.5433 ± 0.0155
DE	EI	$\mathbf{0.0011 \pm 0.0001}$	$\mathbf{0.9031 \pm 0.0436}$	0.0363 ± 0.0010	-0.2927 ± 0.0096
RF	EI	0.0043 ± 0.0003	1.0925 ± 0.0459	0.0146 ± 0.0004	0.8718 ± 0.0761
BNN	EI	0.0332 ± 0.0018	4.8430 ± 0.2239	0.1052 ± 0.0007	0.7928 ± 0.0136
GP	UCB	0.0026 ± 0.0002	1.5156 ± 0.0560	0.0149 ± 0.0004	-0.5297 ± 0.0154
DE	UCB	0.0012 ± 0.0001	0.9159 ± 0.0437	0.0369 ± 0.0009	-0.2862 ± 0.0098
RF	UCB	0.0043 ± 0.0002	1.0979 ± 0.0455	0.0157 ± 0.0004	0.9205 ± 0.0779
BNN	UCB	0.0104 ± 0.0007	2.6292 ± 0.1176	0.1013 ± 0.0006	1.0458 ± 0.0088
GP	TS	0.0046 ± 0.0003	1.7544 ± 0.0643	0.0125 ± 0.0003	-0.5814 ± 0.0173
DE	TS	0.0016 ± 0.0002	1.0321 ± 0.0489	0.0364 ± 0.0009	-0.2522 ± 0.0100
RF	TS	0.0017 ± 0.0002	1.3192 ± 0.0497	$\mathbf{0.0101 \pm 0.0002}$	0.8893 ± 0.0859
BNN	TS	0.0176 ± 0.0009	2.9900 ± 0.1231	0.1025 ± 0.0005	1.0644 ± 0.0091
GP	AVG	0.0034 ± 0.0001	1.6025 ± 0.0342	0.0142 ± 0.0002	-0.5515 ± 0.0093
DE	AVG	$\mathbf{0.0013 \pm 0.0001}$	$\mathbf{0.9504 \pm 0.0263}$	0.0365 ± 0.0005	-0.2770 ± 0.0057
RF	AVG	0.0034 ± 0.0001	1.1699 ± 0.0273	$\mathbf{0.0135 \pm 0.0002}$	0.8939 ± 0.0462
BNN	AVG	0.0204 ± 0.0007	3.4874 ± 0.0965	0.1030 ± 0.0003	0.9676 ± 0.0069
GP (recal.)	AVG	0.0060 ± 0.0002	1.8416 ± 0.0400	0.0149 ± 0.0002	-0.6552 ± 0.0058
DE (recal.)	AVG	$\mathbf{0.0019 \pm 0.0001}$	$\mathbf{1.1468 \pm 0.0320}$	0.0418 ± 0.0005	-0.3123 ± 0.0042
RF (recal.)	AVG	0.0029 ± 0.0001	1.1907 ± 0.0292	$\mathbf{0.0112 \pm 0.0001}$	-0.5700 ± 0.0047
BNN (recal.)	AVG	0.0383 ± 0.0013	4.9472 ± 0.1458	0.0937 ± 0.0003	0.7728 ± 0.0136

tasks, the model calibration can be assessed using the expected calibration error

$$\text{ECE} = \sum_p w_p (C_y(p) - p)^2, \tag{5}$$

where $C_y(p)$ is defined as

$$C_y(p) = \frac{1}{N_T} \sum_{t=1}^{N_T} \mathbb{I}[y_t \leq F_t^{-1}(p)], \tag{6}$$

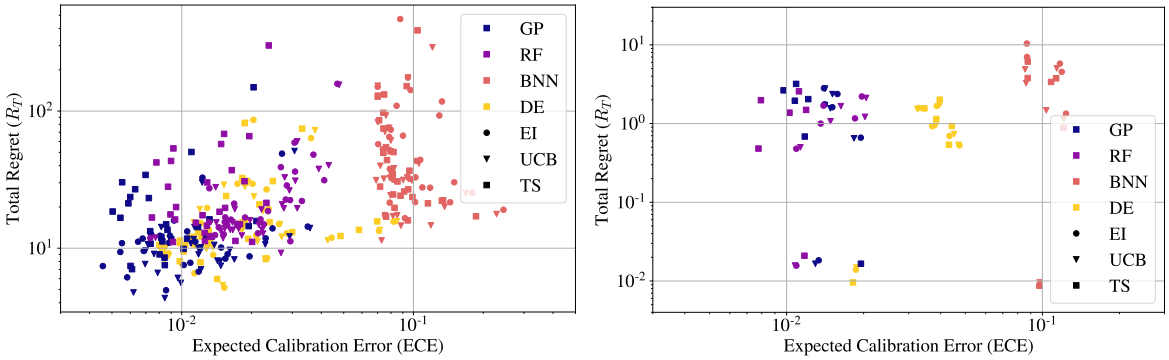
where F_t^{-1} is the quantile function, i.e. $F_t^{-1}(p) \equiv \inf_y \{y \,|\, p \leq F_t(y)\}$, for the t'th datapoint evaluated at percentile p, $\mathbb{I}$ is an indicator function and w_p can be chosen to adjust the importance of percentiles with fewer datapoints. Throughout this paper, we assume $w_p = 1 \; \forall \, p$. The closer the ECE is to zero, the better calibrated the model is.

Recalibration Kuleshov et al. [2018] also proposes a general procedure for recalibrating any model. A so-called recalibrator model C is constructed using an independent and identically distributed (i.i.d.) validation set and subsequently, applied to adjust the CDF of the model's predictive distribution F_t for some observation y_t, i.e. the recalibrated predictive distribution is $C \circ F_t$. This is done via learning an isotonic mapping: $C : [0, 1] \rightarrow [0, 1]$ from the predicted probabilities of events of the form $(-\infty, y_t]$ to the corresponding empirical probabilities. In Deshpande and Kuleshov [2021], a recalibration method for BO specifically is proposed, in which the recalibrator model is learnt via leave-one-out

CV on the samples gathered during BO. After training the recalibrator model C, the relevant summary statistics (e.g. moments and intervals) of the recalibrated distributions can be computed numerically from $C \circ F_t$. See Alg. 1 in Kuleshov et al. [2018] for more details.

3 Experiments

In this section, we describe a collection of numerical experiments designed to study and investigate the relationship between calibration and regret. We focus our study on four popular models, namely GPs, RFs, DEs, and BNNs. For GPs, DEs, and BNNs, we assume an isotropic Gaussian likelihood and for RFs, we impose a Gaussian predictive distribution, where the mean and variance are estimated as the sample mean and variance of the tree predictions. Our experiments are based on both synthetic and real-world data: for experiments with synthetic data, we use a number of problems from the common benchmark suites for optimization called Sigopt [Jamil and Yang, 2013, Dewancker et al., 2016], and for the real-world data, we apply BO to hyperparameter tuning of various machine learning models including feed-forward Neural Networks, Convolutional Neural Networks and SVMs used on on or more datasets such as MNIST [Lecun et al., 1998], Fashion-MNIST [Xiao et al., 2017], AG News classification [Zhang et al., 2015] and Wine classification [Dua and Graff, 2017]. For all experimental details, see Supplementary Material.



(a) Test calibration vs regret of synthetic data experiments.

(b) Test calibration vs regret of real data experiments

Figure 1: Total Regret vs. ECE for synthetic data experiments and hyperparameter tuning experiments. The colors in the scatter plot indicate the type of surrogate model, and the marker indicates the AF used. **OBS:** Each point in the scatter plots corresponds to an average of 100 random seeds for each specific configuration of the experiment.

Experimental Setup In the synthetic setting, we perform BO experiments on a total of 128 optimisation problems spanning input dimensions ($D \in \{1, 2, .., 10\}$) from the Sigopt benchmark. For each optimisation problem, we repeat the experiment 100 times using different random initialization of both the BO routines and seeds. We do this for all combinations of surrogates and AFs, of which we use the previously mentioned EI, UCB and TS. We consistently use ten initial i.i.d. random samples followed by 90 BO iterations for all experiments. We add Gaussian distributed noise to all Sigopt objective functions. For reference, we also include a random sampling (RS) acquisition function. In the hyperparameter tuning setting, we perform BO experiments on a total of 6 different hyperparameter tuning problems. The surrogate models and AFs are the same as in the synthetic setting, and we similarly sample 10 i.i.d. points to initiate the BO session, and then run 90 BO iterations.

Our key performance metrics are regret, calibration error, sharpness as defined in the following. We report the calibration error, ECE as being the mean squared calibration error evaluated on a large i.i.d. test set ($N_{\text{test}} = 5000$) as

$$\text{ECE} = \frac{1}{P} \sum_{j=1}^{P} (C_y(p_j) - p_j)^2, \tag{7}$$

where $C_y(p_j)$ is defined in eq. (6) and for $0 \leq p_1 \leq p_2 ... \leq p_P \leq 1$ as suggested by Kuleshov et al. [2018]. We use $P = 20$ with equidistant p_j values. We quantify the BO performance using the regret metric, where we define the instantaneous regret for the last iteration T as follows

$$\mathcal{R}_I = y_{\text{min}} - y(x_T^*), \tag{8}$$

where $y(x)$ is the objective function value obtained at point x, $y_{\text{min}} \equiv \min_x y(x)$ is function value at the global minimum, and $x_T^* \equiv \arg\min_{x_t} \{y(x_t)\}_{t=1}^T$ is the input value for the best observation after T iterations. Similarly, the total regret is the sum of the instantaneous regret across all iterations

$$\mathcal{R}_T = \sum_{i=1}^{T} \left[y_{\text{min}} - y(x_i^*) \right]. \tag{9}$$

All regret values are reported after standardizing objective function values. Finally, we report the sharpness as the average negative entropy of the predictive distributions across all BO iterations. For the choices of surrogate models, we use a GP with an RBF kernel, and optimize hyperparameters of the kernel at every BO iteration using the exact marginal likelihood [Rasmussen, 2003]. The mean-field BNN has a single hidden layer with 10 hidden neurons, which is trained using the ELBO loss [Blei et al., 2018]. The DEs consists of 10 neural networks with two hidden layers and are all trained using the MSE loss and Adam optimiser [Kingma and Ba, 2014]. Finally, the RFs have their hyperparameters tuned via CV on a grid of hyperparameters at each BO iteration. With regards to the AFs, we use EI as defined in Eq. 2, UCB with $\beta = 1$, and only sample one posterior function at each BO step when using TS. See detailed experimental details and descriptions in the Supplementary Material. Code will be released on GitHub along with the camera-ready version.

Experiment results The results for the synthetic and real data experiments are summarized in Tables 1 and 2, respectively. We observe that in the synthetic setting, GPs outperform all other models both in terms of instantaneous regret and more importantly, total regret, although closely followed by DEs. RFs perform relatively well (at all times better than random sampling), whilst the BNNs exhibit poor performance and are often outperformed by random sampling. Finally, we see that the GP is best calibrated overall, and that all surrogate models have their lowest ECE when random sampling is used. This is overall not surprising as the ECE is evaluated on a large i.i.d. test set, which is more well-represented by i.i.d. training samples compared to strongly dependent samples acquired iteratively through BO. For the real-data experiments in Table 2, we see that DEs outperform all other models in terms of both regret types, and are closely followed by both GPs and RFs which perform comparatively. Once again, GPs are the best calibrated when random sampling is employed.

Relationship between calibration and regret In order to investigate the relationship between BO performance (regret) and calibration (ECE), we first compute the Pearson correlation coefficient between the total regret values and the ECE values, which yield strong and statistically significant coefficients of 0.33 and 0.43 for synthetic and hyperparameter tuning experiments, respectively (see Table 3). The strong positive association is also visually confirmed by the scatter plots in Figures 1. It is also evident from these plots that the type of surrogate model is important for both ECE and total regret. Therefore, we also compute the partial correlation coefficient controlling for the model type yielding -0.03 and -0.22 for synthetic and real data, respectively. Interestingly, both correlations become weaker and statistically insignificant (at level $\alpha = 0.05$) leading to an instance of Simpson's paradox [Wagner, 1982]. To further investigate this, we conducted a multiple linear regression analysis for total regret vs ECE controlling for both the type of model and the specific problem instance. The results showed that both the common slope and model-specific slopes for ECE were generally weak and statistically insignificant (see all details in the Supplementary Material). In summary, these results show that models with high ECE are generally associated with high regrets, however, this association diminishes when we control for the type of surrogate model. To further scrutinize these observations, we conduct two additional experiments: an intervention study and a recalibration study.

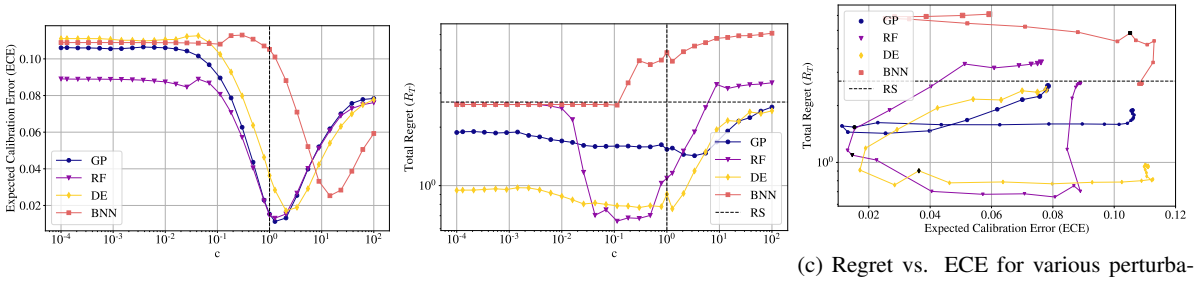
Table 3: Correlation values between regret and ECE.

	Synth. Data	**Real Data**
Correlation	0.33 ($p < 10^{-9}$)	0.43 ($p = 0.001$)
Partial Correlation — Model	-0.03 ($p = 0.59$)	-0.22 ($p = 0.076$)

Intervention study: Perturbing Predictive Uncertainties In the intervention study, we explicitly manipulate calibration by perturbing the predictive uncertainty of each model during the BO protocol. Specifically, we multiply the standard deviation of the posterior distribution for all models by a constant $c \in \left[10^{-4}, 10^{2}\right]$ and observe the resulting effect on ECE and total regret. We conduct this experiment for the 6 different hyperparameter tuning problems using the EI acquisition function and repeat the experiment with 40 different seeds. In Figure 2 we show the calibration error (a) and total regret (b) as a function of the multiplicative constant c. Several interesting observations can be

made from Figure 2. First, all models exhibit the smallest calibration error at $c > 1$, which indicates some degree of overconfidence, and thus, increasing the predictive variance slightly generally improves calibration. Interestingly, DEs and GPs are somewhat robust to these perturbations in their predictive uncertainties with regard to regret, while RFs even seem to benefit from having the uncertainties reduced. Finally, in panel (c) we plot regret vs calibration error for each value of c, where each marker is scaled with the size of c and $c = 1$ is marked with black. We have connected the dots for each surrogate function, going from smallest to largest c. From this plot, we observe that perturbing by $c > 1$ rapidly increase both regret and ECE, but perturbations with $c < 1$ are less harmful and may actually lead to improved performance. In other words, the results from this experiment suggest that miscalibration caused by models being generally underconfident, i.e. $c > 1$, is more detrimental to BO performance compared to models being overconfident, i.e. $c < 1$.

Recalibration study: Recalibration during BO In the recalibration study, we investigate whether recalibrating the models during the BO protocol improves BO performance following the recalibration procedure proposed by Deshpande and Kuleshov [2021]. We do this by re-running our BO experiments on real data, where we use leave-one out CV on the training data obtained during BO to learn a recalibration model and adjust the resulting predictive distributions accordingly. The results can be seen in the last section in the bottom of Table 2. Except for RFs, it is seen that both regret and ECE are generally worse *after recalbration*. This may seem counter-intuitive, but then recall that we compute the recalibration model using leave-one-out on the training set, but we measure the expected calibration on an independent test set. The recalibration procedure may have improved the calibration metric on the training dataset, but in our experiments, it does not generalize to an independent test set. We note that RFs do benefit from recalibration, but this might be explained by the fact that sharpness is substantially reduced after recalibration. We will shed more light on these observations in the next section.



(a) ECE after BO protocol

(b) Regret after BO protocol

(c) Regret vs. ECE for various perturbations.

Figure 2: The effect on test calibration and regret when disturbing the posterior predictive uncertainty by $c \cdot \sigma(\mathbf{x})$ during the BO protocol. (a) Shows the overall ECE of each model when a perturbation of $c \cdot \sigma(\mathbf{x})$ is done in each iteration, (b) shows the corresponding total regret, and (c) depicts how regret and calibration varies together for the same experiments. The size of the markers here indicate how large c is, and the plot lines go from smallest to largest c. Black points are when $c = 1$.

4 Discussion and Summary

In the previous section, we described and performed a number of numerical experiments to analyze the relationship between calibration and regret for BO. In this section, we will summarize and discuss some of the key take-aways as well as expand the analysis with a theoretical perspective.

Take-away 1: Gaussian Processes and Deep Ensembles work well for BO under most conditions. Our results for synthetic data is consistent with the apparent consensus that GPs are strong surrogates for BO and that they outperform the competing methods in terms of regret (both total and instant) (see Table 1), with DEs being close followers. Surprisingly, in the hyperparameter tuning experiments, DEs perform exceedingly well, with RFs and GPs performing equally well. One should however note the practical concern that DEs is computationally more expensive to train during the BO procedure, but that this could be rationalized if such compute time is cheap relative to querying the objective function. In both experiments, the mean-field BNNs perform significantly worse than all other methods, including random search. Similar behavior has also been observed in other experimental design settings, e.g. active learning [Foong et al., 2020]. In terms of ECE, the GPs performed slightly better than the RFs and DEs in the synthetic setting, whilst RFs and GPs perform comparably in the hyperparameter tuning setting. Again, we notice that the

mean-field BNNs are inferior to the other methods in both experiments.

Take-away 2: Correlation between BO performance and calibration diminishes when controlling for the type of surrogate model. For the synthetic and hyperparameter experiments, our analysis showed strong positive correlations 0.330 and 0.434, respectively, between total regret and ECE, when computed across all problems, seeds, acquisition functions, and surrogate models. However, when we control for the type of surrogate model, the correlation becomes much weaker and statistically insignificant (see Table 3). That is, within each model family, BO trials with lower calibration errors are generally not linearly associated with lower regret and in turn better BO performance.

Take-away 3: Under-confidence might be more harmful to BO compared to overconfidence. In our intervention study, we manipulated all surrogate models to be either under- or overconfident during the BO protocol by multiplying their predictive uncertainties by a constant $c > 0$, where $0 < c < 1$ implies more confident predictions, and $c > 1$ implies less confident predictions. The results showed that all models exhibited some degree of overconfidence, which may not be surprising. However, the results also showed that BO performance decreased (i.e. regret increased) rapidly for all models for $c > 1$, whereas BO performance was much more robust to perturbations with $c < 1$, which actually caused an increase in BO performance in some cases. Only for the GP, we observed a slight temporary improvement in regret for $c > 1$. It is also worth emphasizing that the value of c leading to optimal calibration did not coincide with the values for optimal regret. Finally, it is evident from eq. (2) that changing c also affects the effective exploitation-exploration trade-off which, in turn, may also impact the regret (the optimal trade-off is also likely to be intrinsic to the optimisation problem). This can be observed in Figure 2, where both very small and very large values of c caused the methods to behave more like random search.

Take-away 4: Recalibration does generally not improve BO performance. We further investigated the potential benefit of recalibrating the surrogate models during the BO process using a leave-one-out procedure. However, in our recalibration experiments on the hyperparameter tuning datasets, the recalibration procedure only lead to improved ECE (measured on a proper independent test set) for two surrogate models, namely the BNNs and the RFs. In the other cases, it actually worsened the ECE. Moreover, we also noticed that all models got worse total regret performance after employing the recalibration procedure.

Hypothesis: Calibration curves are not reliable for small sample sizes. Recent work by Deshpande and Kuleshov [2021] observed that re-calibration might aid BO by yielding smaller total regret in some trials and smaller instant regret for the BO last iteration in fewer trials. However, our experiments suggest that recalibration might actually degrade BO performance. Kuleshov et al. [2018] state that a sufficiently large i.i.d. validation set is a required condition for successful recalibration, which is in stark contrast to the sample collection during BO which is not i.i.d. due to the inherent sequential nature of BO algorithms and is often characterized by small sample sizes.

To investigate this hypothesis, our starting point will be a simple regression setting, where $p_y(y|x)$ denotes the true data generating distribution of y given an input x. We further assume a trained model with predictive distribution $p_t(y|x)$ aiming to mimic p_y via training samples. Consider now the task of assessing the calibration of model using a set of i.i.d. validation samples $\{y_1, y_2, ..., y_N\}$. Given the small sample sizes typically used in BO, a natural question to ask is how accurately can we assess the calibration curve as a function of the size of the validation set N? To answer this question, we consider the variance of the estimator in eq. (6) and analyze its decay rate as a function of the sample size N. The result is summarized in the following statement:

Proposition 1. *Let F_i be the CDF of the predictive distribution for the i'th observation and let $\{y_i\}_{i=1}^N$ be i.i.d. samples $y_i \sim p_y$. For $\mathcal{C}_y(p) = \frac{1}{N}\sum_{i=1}^N \mathbb{I}\left[y_i \leq F_i^{-1}(p)\right]$, then the variance of $C_y(p)$ decays as $\mathbb{V}\left[\mathcal{C}_y(p)\right] = \mathcal{O}(N^{-1})$.*

Proof. Let $\mathcal{C}_y(p) = \frac{1}{N}\sum_{i=1}^N z_i$ for $z_i \equiv \mathbb{I}\left[y_i \leq F_i^{-1}(p)\right]$. The variance of $\mathcal{C}_y(p)$ is then given by

$$\mathbb{V}\left[\mathcal{C}_y(p)\right] = \mathbb{V}\left[\frac{1}{N}\sum_{i=1}^N z_i\right]$$

by independence each z_i, and

$$\mathbb{V}\left[\mathcal{C}_y(p)\right] \leq \frac{1}{N^2}\sum_{i=1}^N \sup_i \mathbb{V}\left[z_i\right] = \frac{1}{N^2}\sum_{i=1}^N \frac{1}{2^2} = \frac{1}{N}\frac{1}{2^2}.$$

Hence, it follows the variance of $\mathcal{C}_y(p)$ is bounded by

$$\mathbb{V}\left[\mathcal{C}_y(p)\right] \leq \mathcal{O}\left(N^{-1}\right). \tag{10}$$

See Supplementary Material for detailed proof. □

We also confirmed this result empirically and observe results perfectly consistent with the predictions from Proposition 1 (see in the Supplementary Material), i.e. the maximum standard deviation of the estimator for $C_y(p)$ decays as $\frac{1}{\sqrt{N}}$. Next, we assume our model is perfect, i.e. $p_t(y|x) = p_y(y|x)$, and ask what is the ECE caused by a small sample size alone. The results are summarized in the next two statements:

Proposition 2. *Let F_i be the CDF of the predictive distribution perfect model, i.e. $p_t(y|x) = p(y|x)$. If F_i is strictly monotonic, it holds that $\mathbb{V}\left[\mathcal{C}_y(p)\right] = \frac{p(1-p)}{N}$ for all p.*

Proof. In this setting, we have

$$z_i = \mathbb{I}\left[y_i \leq F_i^{-1}(p)\right] = \mathbb{I}\left[F_i(y_i) \leq p\right] = \mathbb{I}\left[u_i \leq p\right],$$

where $u_i \sim \mathcal{U}\left[0,1\right]$ are uniformly distributed on the unit interval due to the probability integral transform. Since $\{u_i\}_{i=1}^N$ are also independent, it follows that $S_n = \sum_{i=1}^N z_i \sim \text{Binomial}(N,p)$. Therefore, we have

$$\mathbb{V}\left[\mathcal{C}_y(p)\right] = \mathbb{V}\left[N^{-1}S_N\right] = N^{-2}\mathbb{V}\left[S_N\right] = N^{-1}p(1-p).$$

This completes the proof. □

Proposition 3. *Let ECE $= \sum_{j=1}^P w_j(p_j - \mathcal{C}_y(p_j))^2$ be the weighted mean square calibration error. Assume $w_i \in [0,1]$ and $0 < p_1 < p_2 < ... < p_P < 1$ are fixed, and assume the CDF of the predictive distribution is equal to the true data distribution (almost everywhere), then it holds that $\mathbb{E}\left[ECE\right] = \mathbb{E}\left[\frac{1}{P}\sum_{j=1}^P w_j p_j(1-p_j)\right] = \mathcal{O}(N^{-1})$.*

Proof. See Supplementary Material. □

Take-away 5: Calibration curves are not reliable for small sample sizes Proposition 1 and Proposition 2 state that the variance of the estimator of the empirical calibration decreases with $\mathcal{O}\left(N^{-1}\right)$. This implies that empirical calibration curves are likely to be unreliable for small sample sizes and to improve the accuracy of the estimates by one decimal point, one needs to increase the size of the validation set by a factor of 100, which will often be infeasible in practical BO settings. Furthermore, Proposition 3 states that even for a perfect model, the expected ECE is $\mathcal{O}\left(N^{-1}\right)$. Therefore, for small sample sizes, one should be careful when concluding that a model is mis-calibrated, since the observed ECE might as well be caused by the sample size. Even worse, when performing recalibration in this scenario, one might risk adjusting the model in the "wrong direction" causing the model to be more miscalibrated than the original model. In the supplementary material, we show several examples of this phenomenon. Although our empirical and theoretical analysis are focused on the i.i.d. setting, we expect the effect to be even more severe in the non-i.i.d. case since the effective sample size is typically smaller for correlated samples [Thiébaux and Zwiers, 1984]. Therefore, we claim that these effects may have profound impact on recalibration in BO protocols.

Future work Our study indicates that the common way to diagnose calibration (on a large test set) might not be sensible for BO and that future studies about calibration metrics more relevant to BO are needed. It will also be of great interest to explore the relationship between calibration and regret from a casual perspective. Lastly, it is interesting to dig deeper into the effects of under- vs. over-confidence on BO performance.

References

A. Agnihotri and N. Batra. Exploring bayesian optimization. *Distill*, 5(5):e26, 2020.

S. Belakaria, A. Deshwal, N. K. Jayakodi, and J. R. Doppa. Uncertainty-aware search framework for multi-objective bayesian optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10044–10052, 2020.

J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. 2018.

N. Bliznyuk, D. Ruppert, C. Shoemaker, R. Regis, S. Wild, and P. Mugunthan. Bayesian calibration and uncertainty analysis for computationally expensive models using optimization and radial basis function approximation. *Journal of Computational and Graphical Statistics*, 17(2):270–294, 2008.

J. Busk, P. B. Jørgensen, A. Bhowmik, M. N. Schmidt, O. Winther, and T. Vegge. Calibrated uncertainty for molecular property prediction using ensembles of message passing neural networks. *Machine Learning: Science and Technology*, 3(1):015012, 2021.

Y. Chung, I. Char, H. Guo, J. Schneider, and W. Neiswanger. Uncertainty toolbox: an open-source library for assessing, visualizing, and improving uncertainty quantification. *arXiv preprint arXiv:2109.10254*, 2021.

S. Deshpande and V. Kuleshov. Calibration improves bayesian optimization. *arXiv preprint arXiv:2112.04620*, 2021.

A. Deshwal, S. Belakaria, and J. R. Doppa. Bayesian optimization over hybrid spaces. *arXiv preprint arXiv:2106.04682*, 2021.

I. Dewancker, M. McCourt, S. Clark, P. Hayes, A. Johnson, and G. Ke. A stratified analysis of bayesian optimization methods. *arXiv preprint arXiv:1603.09441*, 2016.

D. Dua and C. Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

A. N. Elmachtoub, R. McNellis, S. Oh, and M. Petrik. A practical method for solving contextual bandit problems using decision trees. *CoRR*, abs/1706.04687, 2017. URL `http://arxiv.org/abs/1706.04687`.

A. Foong, D. Burt, Y. Li, and R. Turner. On the expressiveness of approximate inference in bayesian neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15897–15908. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/b6dfd41875bc090bd31d0b1740eb5b1b-Paper.pdf`.

Z. Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452–459, 2015.

M. Jamil and X.-S. Yang. A literature survey of benchmark functions for global optimization problems. *arXiv preprint arXiv:1308.4008*, 2013.

D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive Black-Box functions. *J. Global Optimiz.*, 13(4):455–492, Dec. 1998.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014.

V. Kuleshov and S. Deshpande. Calibrated and sharp uncertainties in deep learning via density estimation. In *International Conference on Machine Learning*, pages 11683–11693. PMLR, 2022.

V. Kuleshov, N. Fenner, and S. Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning*, pages 2796–2804. PMLR, 2018.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Z. Nado, N. Band, M. Collier, J. Djolonga, M. W. Dusenberry, S. Farquhar, A. Filos, M. Havasi, R. Jenatton, G. Jerfel, et al. Uncertainty baselines: Benchmarks for uncertainty & robustness in deep learning. *arXiv preprint arXiv:2106.04015*, 2021.

Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. V. Dillon, B. Lakshminarayanan, and J. Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*, 2019.

C. E. Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

H. Song, T. Diethe, M. Kull, and P. Flach. Distribution calibration for regression. In *International Conference on Machine Learning*, pages 5897–5906. PMLR, 2019.

J. T. Springenberg, A. Klein, S. Falkner, and F. Hutter. Bayesian optimization with robust bayesian neural networks. *Advances in neural information processing systems*, 29:4134–4142, 2016.

N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, may 2012. doi: 10.1109/tit.2011.2182033. URL `https://doi.org/10.1109%2Ftit.2011.2182033`.

H. J. Thiébaux and F. W. Zwiers. The interpretation and estimation of effective sample size. *Journal of Applied Meteorology and Climatology*, 23(5):800–811, 1984.

W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25:285–294, 1933.

R. Turner, D. Eriksson, M. McCourt, J. Kiili, E. Laaksonen, Z. Xu, and I. Guyon. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In *NeurIPS 2020 Competition and Demonstration Track*, pages 3–26. PMLR, 2021.

C. H. Wagner. Simpson's paradox in real life. *The American Statistician*, 36:46–48, 1982.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. URL `https://arxiv.org/abs/1708.07747`.

X. Zhang, J. J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. *CoRR*, abs/1509.01626, 2015. URL `http://arxiv.org/abs/1509.01626`.

SUPPLEMENTARY MATERIAL

Hyperparameter Tuning Datasets

When collecting our hyperparameter tuning datasets, the combinations of models and datasets are as follows:

Table 4: Model and Data Combinations for Hyperparameter Tuning

	MNIST	FashionMNIST	AG News Classification	Wine Classification
FFNN	✓	✓	✓	
CNN	✓	✓		
SVM				✓

For each of the models we then select a number of hyperparameters which we want to tune, create a grid for these hyperparameters and train a model for each of these hyperparameter sets (the BO input is thus hyperparameters and the output is validation performance). The FFNN simply has a single hidden layer with a ReLU activation function and a single dropout layer, except in the case of the AG News Classification where the "hidden layer" is an embedding layer using the `nn.EmbeddingBag` from `torch`. The CNN is a network with two convolution layers with kernel size $(5, 5)$ of output channels 16 and 32 respectively, and a single hidden and dropout layer. Max pooling is also used with a kernel size of $(2, 2)$ at every convolution layer. The SVM used is the `sklearn.SVM.SVC` from `sklearn`. The hyperparameters and their grid specification can be seen here:

Table 5: Grid Specifications for Hyperparameter Tuning

	Training Epochs	**Dropout Rate**	**Learning Rate (log space)**	**Batch Size Train**	**Hidden Size**	**C (log space)**	**γ (log space)**
FFNN	np.linspace(1, 10, 10)	np.linspace(0, 0.8, 10)	np.linspace(-11.51, -2.23, 10)	np.arange(8, 256, 32)	np.linspace(1, 271, 10)		
CNN	np.linspace(1, 10, 10)	np.linspace(0, 0.8, 10)	np.linspace(-11.51, -2.23, 10)	np.arange(8, 256, 32)	np.linspace(1, 271, 10)		
SVM						np.linspace(-6.9, 4.6, 100)	np.linspace(-11.51, -2.23, 100)

Mathematical Proofs

Proposition 1: Let F_i be the CDF of the predictive distribution for the i'th observation and let $\{y_i\}_{i=1}^n$ be i.i.d. samples $y_i \sim p_y$. For $\mathcal{C}_y(p) = \frac{1}{n}\sum_{i=1}^n \mathbb{I}\left[y_i \leq F_i^{-1}(p)\right]$, then the variance of $C_y(p)$ is bounded by $1/n$, i.e. $\mathbb{V}\left[C\right] = \mathcal{O}(n^{-1})$.

Proof: First, we show that the variance is bounded by $\mathcal{O}(n^{-1})$. We have

$$\mathcal{C}_y(p) = \frac{1}{n}\sum_{i=1}^n \mathbb{I}\left[y_i \leq F_i^{-1}(p)\right] = \frac{1}{n}\sum_{i=1}^n z_i, \tag{11}$$

where $z_i \equiv \mathbb{I}\left[y_i \leq F_i^{-1}(p)\right]$. The variance of $\mathcal{C}_y(p)$ is then by give

$$
\begin{aligned}
\mathbb{V}\left[\mathcal{C}_y(p)\right] &= \mathbb{V}\left[\frac{1}{n}\sum_{i=1}^{n} z_i\right] \\
&= \frac{1}{n^2}\mathbb{V}\left[\sum_{i=1}^{n} z_i\right] \\
&= \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{V}\left[z_i\right] \\
&\leq \frac{1}{n^2}\sum_{i=1}^{n}\sup_i \mathbb{V}\left[z_i\right] \\
&\leq \frac{1}{n^2}\sum_{i=1}^{n}\frac{1}{2^2} \\
&= \frac{1}{n}\frac{1}{2^2}
\end{aligned}
\tag{12}
$$

Hence, it also follows the standard deviation of $\mathcal{C}_y(p)$ is bounded by

$$
\sqrt{\mathcal{C}_y(p)} \leq \sqrt{\frac{1}{n}\frac{1}{2^2}} = \frac{1}{2\sqrt{n}} = \mathcal{O}\left(\frac{1}{\sqrt{n}}\right). \tag{13}
$$

This completes the proof of the first statement.

Lemma 1: Given a perfectly calibrated model, it holds that $\mathbb{V}\left[\mathcal{C}_y(p)\right] = \frac{p(1-p)}{n}$ for all p.

Proof: In this setting, we have

$$
z_i = \mathbb{I}\left[y_i \leq F_i^{-1}(p)\right] = \mathbb{I}\left[F_i(y_i) \leq p\right] = \mathbb{I}\left[u_i\right] \leq p\right], \tag{14}
$$

where $u_i \sim \mathcal{U}\left[0,1\right]$ are uniformly distributed on the unit interval due to the probability integral transform. Since $\{u_i\}_{i=1}^n$ are also independent, it follows that

$$
S_n = \sum_{i=1}^{n} z_i \sim \text{Binomial}(n,p). \tag{15}
$$

Therefore, it follows that

$$
\begin{aligned}
\mathbb{V}\left[\mathcal{C}_y(p)\right] &= \mathbb{V}\left[\frac{1}{n}S\right] = \frac{1}{n^2}\mathbb{V}\left[S\right] \\
&= \frac{1}{n^2}np(1-p) = \frac{p(1-p)}{n}.
\end{aligned}
\tag{16}
$$

This completes the proof.

Proposition 2: Let $E_c = \sum_{j=1}^{m} w_j(p_j - \mathcal{C}_y(p_j))^2$ be the weighted mean square calibration error. Assume $w_i \in [0,1]$ and $0 < p_1 < p_2 < ... < p_m < 1$ are fixed, and assume the CDF of the predictive distribution is equal to the true data distribution (almost everywhere), then it holds that $\mathbb{E}\left[E_c\right] = \frac{1}{n}\sum_{j=1}^{m} w_j p_j(1-p_j) = \mathcal{O}(n^{-1})$ if $y_i \sim p_y$ are i.i.d. samples.

The calibration error E_C is defined as follows

$$
E_c = \sum_{j=1}^{m} w_j(p_j - \mathcal{C}_y(p_j))^2, \tag{17}
$$

where each $w_i \in [0,1]$ is a weight and $0 \leq p_1 < p_2 < ... < p_m < 1$ is predefined set of points.

In order to compute the expectation of E_C, we first expand:

$$E = \sum_{j=1}^{m} w_j (p_j^2 + \mathcal{C}_y(p_j)^2 - 2p_j\mathcal{C}_y(p_j)) \tag{18}$$

$$= \sum_{j=1}^{m} w_j \mathcal{C}_y(p_j)^2 - 2\sum_{j=1}^{m} w_j p_j \mathcal{C}_y(p_j)) \tag{19}$$

Then it follows that

$$\mathbb{E}_{\mathbb{C}}[E] = \mathbb{E}\left[\sum_{j=1}^{m} w_j p_j^2 + \sum_{j=1}^{m} w_j \mathcal{C}_y(p_j)^2 - 2\sum_{j=1}^{m} w_j p_j \mathcal{C}_y(p_j))\right] \tag{20}$$

$$= \sum_{j=1}^{m} w_j p_j^2 + \sum_{j=1}^{m} w_j \mathbb{E}\left[\mathcal{C}_y(p_j)^2\right] - 2\sum_{j=1}^{m} w_j p_j \mathbb{E}\left[\mathcal{C}_y(p_j)\right]. \tag{21}$$

The first moment evaluates to

$$\mathbb{E}[C_y(p)] = \int_{-\infty}^{\infty} \mathbb{I}[y_t \leq F_t^{-1}(p)] p_y \mathrm{d}y \tag{22}$$

$$= \int_{-\infty}^{F_t^{-1}(p)} p_y \mathrm{d}y \tag{23}$$

$$= F_y(F_t^{-1}(p)) \tag{24}$$

$$= p. \tag{25}$$

Similarly, the second moment evaluates to

$$\mathbb{E}\left[\mathcal{C}_y(p)^2\right] = \mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n} z_i\right)^2\right] \tag{26}$$

$$= \frac{1}{n^2}\mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{n} z_i z_j\right] \tag{27}$$

$$= \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\left[z_i^2\right] + \frac{1}{n^2}\sum_{j\neq i}\mathbb{E}\left[z_i z_j\right] \tag{28}$$

$$= \frac{1}{n^2}\sum_{i=1}^{n} p + \frac{1}{n^2}\sum_{j\neq i}\mathbb{E}\left[z_i\right]\mathbb{E}\left[z_j\right] \tag{29}$$

$$= \frac{n}{n^2}p + \frac{1}{n^2}\sum_{j\neq i} p^2 \tag{30}$$

$$= \frac{1}{n}p + \frac{1}{n^2}\left(n^2 - n\right)p^2 \tag{31}$$

Rearranging the terms yields

$$\begin{aligned}\mathbb{E}\left[\mathcal{C}_y(p)^2\right] &= \frac{1}{n}p + \frac{n^2-n}{n^2}p^2 \\ &= \frac{1}{n}p - \frac{1}{n}p^2 + p^2 \\ &= \frac{p(1-p)}{n} + p^2\end{aligned} \tag{32}$$

Substituting the moments into eq. (20) yields

$$\begin{aligned}
\mathbb{E}\left[E_C\right] &= \sum_{j=1}^{m} w_j p_j^2 + \sum_{j=1}^{m} w_j \left[\frac{p_j(1-p_j)}{n} + p_j^2\right] \\
&- 2\sum_{j=1}^{m} w_j p_j^2 \\
&= \sum_{j=1}^{m} w_j p_j^2 + \sum_{j=1}^{m} w_j \frac{p_j(1-p_j)}{n} \\
&+ \sum_{j=1}^{m} w_j p_j^2 - 2\sum_{j=1}^{m} w_j p_j^2 \\
&= \frac{1}{n}\sum_{j=1}^{m} w_j p_j (1-p_j) \\
&= \mathcal{O}(n^{-1}).
\end{aligned} \tag{33}$$

This completes the proof.

If p_y and p_t are normal distributions

For non-perfect models we have that $F_y(F_t^{-1}(p)) = g(p)$ where in general $g(p) \neq p$. If both p_y and p_t are normal distributions, the CDF and inverse CDF of a normal are, respectively, given by

$$\begin{aligned}
F(x) &= \frac{1}{2}\left[1 + \text{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right] \\
F^{-1}(p) &= \mu + \sigma\sqrt{2}\text{erf}^{-1}\left(2p-1\right)
\end{aligned}$$

When data comes from $y_t \sim \mathcal{N}(\mu_y, \sigma_y^2)$ and the model is $\mathcal{N}(\mu_t, \sigma_t^2)$, we can write the expectation of the calibration curve as follows

$$\begin{aligned}
g(p) &= F_y(F_t^{-1}(p)) \\
&= \frac{1}{2}\left[1 + \text{erf}\left(\frac{F_t^{-1}(p) - \mu_y}{\sigma_y\sqrt{2}}\right)\right] \\
&= \frac{1}{2}\left[1 + \text{erf}\left(\frac{\mu_t + \sigma_t\sqrt{2}\text{erf}^{-1}\left(2p-1\right) - \mu_y}{\sigma_y\sqrt{2}}\right)\right] \\
&= \frac{1}{2}\left[1 + \text{erf}\left(\frac{\mu_t - \mu_y}{\sigma_y\sqrt{2}} + \frac{\sigma_t}{\sigma_y}\text{erf}^{-1}\left(2p-1\right)\right)\right] \\
&= \frac{1}{2}\left[1 + \text{erf}\left(\frac{\mu_t - \mu_y}{\sigma_y\sqrt{2}} + \frac{\sigma_t}{\sigma_y}\text{erf}^{-1}\left(2p-1\right)\right)\right]
\end{aligned}$$

which also evaluates to p for a perfect model:

$$\begin{aligned}
g(p) &= \frac{1}{2}\left[1 + \text{erf}\left(\frac{0}{\sigma_y\sqrt{2}} + 1 \cdot \text{erf}^{-1}\left(2p-1\right)\right)\right] \\
&= \frac{1}{2}\left[1 + 2p - 1\right] \\
&= p
\end{aligned}$$

Theoretical Calibration Experiment Figures

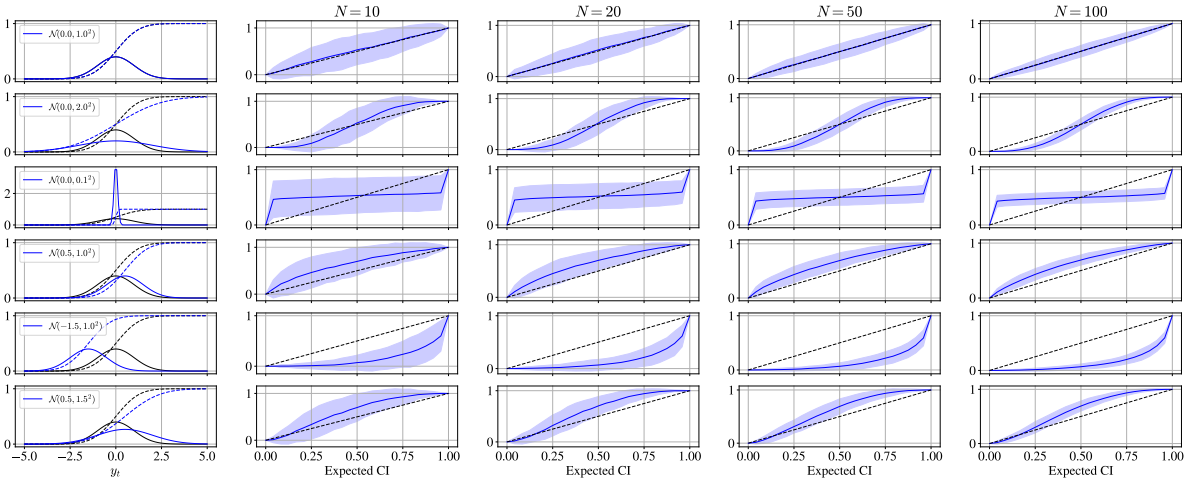


Figure 3: Examples of calibration curves computed on various number of test examples N, when the true data comes from a standard Gaussian and the model (left plots) varies (each row). Even in the best case scenario when samples are i.i.d., a large sample-to-sample variance can be expected in the ranges of N for which BO normally operates. Calibration curve distributions are made from 100 random seeds, and the intervals corresponds to two times the standard deviation.

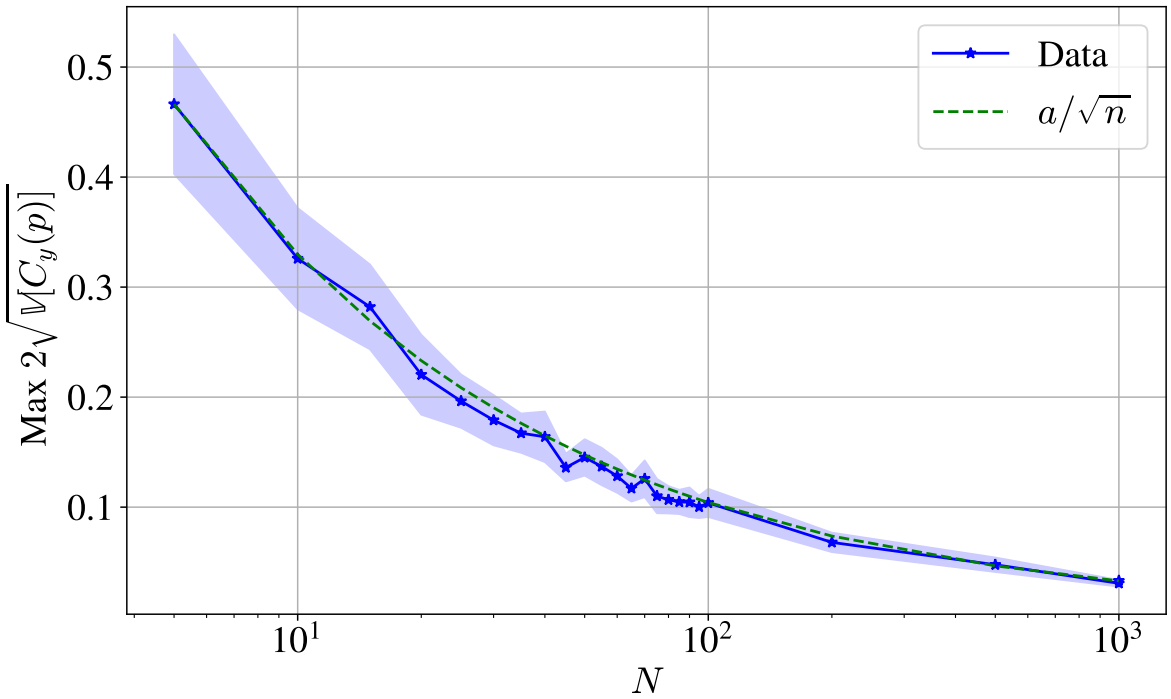


Figure 4: Maximum uncertainty across p for calibration distribution $C_p(y)$ when N samples of y is given for computing the individual calibration curves. We sample 100 models (normal distributions) each with arguments $\mu_i \sim \text{Normal}(0, 1)$ and $\sigma_i \sim \text{LogNormal}(1, 1)$ each modelling data coming from a standard normal. For each experiment 100 calibration curves, that is 100 independent samples of size N from the true model, constitutes the mean and std. We also plot the function $f(N) = a/\sqrt{N}$ for $a \approx 1.05$.

THE END