Quantum methods for machine learning and classical dynamics

Barthe, A.M.

Citation

Barthe, A. M. (2026, March 20). *Quantum methods for machine learning and classical dynamics*. Retrieved from https://hdl.handle.net/1887/4297482

Version:	Publisher's Version
License:	Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from:	https://hdl.handle.net/1887/4297482

Note: To cite this publication please use the final published version (if applicable).

Quantum Methods for Machine Learning and Classical Dynamics

Proefschrift

ter verkrijging van
de graad van doctor aan de Universiteit Leiden,
op gezag van rector magnificus prof. dr. S. de Rijcke,
volgens besluit van het college voor promoties
te verdedigen op vrijdag 20 maart 2026
klokke 13:00 uur

door

Alice Barthe

geboren te Suresnes, Frankrijk
in 1993

Promotor: Prof. dr. V. Dunjko
Co-promotor: Dr. J. Tura

Promotiecommissie: Prof. dr. S. J. van der Molen
Prof. dr. C. W. J. Beenakker
Prof. dr. O. Kiriienko (University of Sheffield)
Dr. Z. Holmes (EPFL Lausanne)
Dr. P. Wallden (University of Edinburgh)

This thesis was funded thanks to the CERN Quantum Technology Initiative and to ESA ϕ-lab Quantum Computing for Earth Observation (QC4EO) initiative.

To those who make me feel at home

Contents

1. Introduction **1**
  1.1. Preface 1
  1.2. Research questions 4
  1.3. List of contributions 8
  1.4. Overview of this thesis 9

2. Background **11**
  2.1. Statistical Learning Theory 11
    2.1.1. Concept classes and PAC learning 11
    2.1.2. Covering numbers 13
  2.2. Quantum Computing 13
    2.2.1. Quantum states 13
    2.2.2. Quantum gates 14
    2.2.3. Measurements 16
  2.3. Parameterized Quantum Circuits 17
    2.3.1. Expressivity 17
    2.3.2. Trainability 21
    2.3.3. Hardness of learning 22
  2.4. Bosonic Hamiltonians and Continuous Variables Quantum Computing 25
    2.4.1. Continuous Variable Quantum Information 25
    2.4.2. Gaussian states 27
  2.5. Complexity Theory and Quantum Classes 28
    2.5.1. Inclusion, hardness, and completeness 29
    2.5.2. The class BQP 29
    2.5.3. The class QMA 30
    2.5.4. The class PostBQP 30

2.5.5. The class DQC1 31

I. Designing PQC-based QML methods 33

3. Gradients and frequency profiles of quantum re-uploading models 35
- 3.1. Introduction 35
- 3.2. Gradients in QRU models 37
  - 3.2.1. Losses for PQC and QRU 37
  - 3.2.2. Gradient of the loss function 38
  - 3.2.3. Absorption Witnesses 40
  - 3.2.4. Gradients in layered QRU models 42
  - 3.2.5. Numerical results 43
- 3.3. Expressivity in QRU models 47
  - 3.3.1. Harmonic representation of quantum states 47
  - 3.3.2. Vanishing high frequencies in QRU models 48
  - 3.3.3. Lipschitz expressivity 52
  - 3.3.4. Extension to generic data generators 54
  - 3.3.5. Numerical results 57
- 3.4. Discussion 60
- 3.5. Conclusions 62
- 3.6. Proofs 64
  - 3.6.1. Proof of Theorem 3.1 64
  - 3.6.2. Proof of Lemma 3.1 67
  - 3.6.3. Details on harmonic representation of QRU models 67
  - 3.6.4. Proof of Theorem 3.2 68
  - 3.6.5. Dirichlet distribution 69
  - 3.6.6. Proof of Corollary 3.3 70
  - 3.6.7. Proof of Theorem 3.3 71
  - 3.6.8. Proof of Theorem 3.4 72
  - 3.6.9. Extension to non-harmonic spectrum 75

4. Parameterized quantum circuits as universal generative models for continuous multivariate distributions 79
- 4.1. Introduction 79
- 4.2. Results 81
  - 4.2.1. Expectation value sampling 81
  - 4.2.2. Universal generative model family 82
  - 4.2.3. Two families of Expectation Value Samplers as Universal Generators 83

- 4.2.4. Necessary conditions for universality 85
- 4.2.5. Trade-off: qubits vs measurements 87
- 4.3. Discussions 87
- 4.4. Methods 90
  - 4.4.1. Random Variable Transformation 90
  - 4.4.2. Universality proofs 92
- 4.5. Appendix 93
  - 4.5.1. Universality Definition 93
  - 4.5.2. Universality Proofs 94
  - 4.5.3. Proof of necessary resources for universality 101
  - 4.5.4. Approximation of expectation values 104
  - 4.5.5. Additional expressivity tools 105

5. Quantum Advantage in Learning Quantum Dynamics 109
- 5.1. Introduction 109
  - 5.1.1. Parameterized Quantum circuits 110
- 5.2. Fourier coefficient extraction algorithm 111
  - 5.2.1. Fourier representation of parameterized circuits . . 111
  - 5.2.2. Fourier representation of expectation values 113
- 5.3. Learning Parameterized Quantum Circuits 114
  - 5.3.1. Concept Class Definition 114
  - 5.3.2. Efficient Quantum Learner 115
  - 5.3.3. Hardness of the learning problem 117
- 5.4. Learning Time Evolution 117
  - 5.4.1. Concept Class Definition 117
  - 5.4.2. Connection between the two concept classes 118
  - 5.4.3. Hardness of the learning problem 119
- 5.5. Beyond log-many parameters 120
  - 5.5.1. Exponentially large spectrum 120
  - 5.5.2. Kernel approach 121
  - 5.5.3. Properties of the feature map 122
- 5.6. Discussion 123
  - 5.6.1. Cardinality of the Concept class 123
  - 5.6.2. Conclusions 124
- 5.7. Appendix 124
  - 5.7.1. LASSO regression 124
  - 5.7.2. Complexity assumption 129
  - 5.7.3. Fourier coefficient extraction algorithm 129
  - 5.7.4. Cardinality of the concept class 137
  - 5.7.5. Alternative Oracle-based algorithms 138

5.7.6. Proof of the PAC learnability of the Hamiltonian dynamics concept class 139
5.7.7. PAC efficient Kernel-based algorithm 140
5.7.8. Noisy Kernel Ridge Regression 141
5.7.9. Conclusion 143
5.7.10. Flipped concept and connection to RFF 144

II. Simulating classical and quantum systems on bosonic systems 145

6. Continuous Variables Quantum Algorithm for solving Ordinary Differential Equations 147
6.1. Introduction 147
6.2. Background 149
6.2.1. Koopman–von Neumann classical mechanics 149
6.2.2. Previous work: Finite Hilbert space approximations 150
6.3. Proposed Algorithm 150
6.3.1. Approximation of the time evolution of the KvN Hamiltonian 150
6.3.2. Chaos and position eigenstates 152
6.3.3. Overall algorithm 153
6.4. Discussion and next steps 154
6.5. Conclusion 155
6.6. Appendix 155
6.6.1. Algorithm subroutines 155
6.6.2. Trotter error and unbounded operators 156

7. The complexity of simulating exponentially large Gaussian bosonic circuits 161
7.1. Introduction 161
7.2. Gate-Based simulation of Gaussian circuits 165
7.2.1. Initialization 165
7.2.2. Evolution 165
7.2.3. Measurements 167
7.3. BQP-completeness of Exponentially large interferometers 168
7.3.1. Problem definition 168
7.3.2. Complexity of the problem 168
7.4. Non-energy-preserving systems 170
7.4.1. Quantum algorithms for dissipative exponentially large differential equations 170

7.4.2. The power of imaginary-time evolution 171
7.4.3. PostBQP-completeness of exponentially large non-energy-conserving systems 173
7.5. Outlook 173
7.6. Appendix 174
7.6.1. Framework 174
7.6.2. From bosonic gates to qubit gates 183
7.6.3. BQP-completeness 190
7.6.4. From unitary quantum circuits to interferometers . 194
7.6.5. Imaginary time evolution is QMA-hard 198
7.6.6. Exponentially large interferometers equipped with squeezing gates are PostBQP-hard 201

8. Conclusion **203**

Bibliography **207**

Acknowledgments **229**

Samenvatting **231**

Summary **233**

Résumé **235**

Curriculum Vitae **237**

CHAPTER 1

Introduction

1.1. Preface

Computation is a critical tool for modern society, enabling everything from scientific discovery to secure communication and artificial intelligence. As our ambitions grow, so do the demands on our *computational resources*, and some problems remain stubbornly out of reach for even the most powerful classical computers. Simultaneously, we have entered a new paradigm where computation is increasingly driven by data feeding into machine learning methods. Today's *machine learning* systems are ubiquitous, from image recognition to language models, and rely on training over vast datasets. This change created new algorithmic challenges and placed intense pressure on hardware and software alike to scale effectively.

Quantum computers have emerged as an avenue to address computational problems that are hard for classical computers by leveraging the principles of *quantum mechanics*. The key to quantum advantage is the ability to exploit quantum phenomena like interference and entanglement. Quantum algorithms can be designed to amplify the probability of correct answers and suppress the wrong ones, manipulating the quantum state as a vector in a high-dimensional space. The operations used to manipulate quantum states are called *quantum gates*, and they can be combined to form *quantum circuits*. Quantum circuits are the quantum analog of classical circuits, and they can be used to perform some computations faster than classical computers. For some problems, such as unstructured search, quantum computers lead to a quadratic speed-up [1]; for others, like factoring [2], the speed-up can be exponential. But quantum computers are not a universal solution to address all computational problems efficiently. As we explore the potential of quantum computing and look for *quantum advantage*, it is crucial to maintain a clear-eyed view of both its strengths and its limitations.

As quantum advantages have been theoretically proven for specific computational problems, the question was extended to whether advantages could also be achieved in *learning tasks* [3]. One attempt at machine learning making use of quantum computers took the form of *variational algorithms* [4]. These are hybrid methods where a quantum circuit, composed of gates governed by a set of adjustable parameters, is used in tandem with a classical optimizer to minimize a cost function, often representative of the prediction error. Unlike traditional quantum algorithms, like Shor's [2] or Grover's [1], which offer provable guarantees under clear assumptions, most variational algorithms are heuristic by nature. They typically come with no formal guarantees about their convergence, accuracy, or advantage over classical methods, but are designed to perform well in practice. In fact, one motivation for variational algorithms was to work within the severe constraints of early quantum hardware [5], where obtaining reliable results is only possible for short circuits.

At the heart of quantum variational algorithms are *parameterized quantum circuits* (PQCs): quantum circuits whose structure is fixed, but which include tunable quantum gates. These gates typically depend on continuous parameters, like rotation angles, that can be optimized to steer the quantum state toward a useful configuration. The idea is simple enough: start with a tunable quantum model, tweak the parameters using feedback from measurements, and hope that this loop uncovers a useful solution. Whether and when this actually leads to a quantum advantage is still an open question, but the framework itself became a foundation for much of today's quantum machine learning research [6–9]. In what follows, we will discuss the challenges and limitations of PQCs in more detail.

Most variational algorithms rely on estimating *expectation values*, real numbers that represent the average outcome of a measurement on a quantum state. In practice, these are not computed directly but estimated statistically by repeatedly measuring the output of a quantum circuit, each time producing a random bitstring. The accuracy of this estimate improves with the number of repetitions (called shots), but only slowly: the standard error decreases as $1/\sqrt{N}$, where N is the number of shots.

While quantum techniques like amplitude amplification [10] can sometimes offer quadratic improvements, the output of a quantum circuit remains fundamentally random. In contrast, classical computing typically allows for deterministic evaluation, and higher accuracies may be reached more easily. Adding a bit in a floating-point number can double the precision of most calculations (except, for example, in Markov chain Monte Carlo methods). This fundamental difference means that achieving high accuracy in variational algorithms often comes at a significant sampling cost.

Another major challenge with PQCs is *trainability*, that is, the ability to find parameter values that solve the problem at hand. The loss function measures how well the quantum model performs on a given task and is typically minimized using gradient-based methods like gradient descent. These methods require computing the derivative of the loss with respect to each parameter, which tells the algorithm in which direction to update the parameters to improve performance. However, many PQC architectures suffer from the so-called *barren plateau* problem [11], where the gradients vanish exponentially with the number of qubits or circuit depth. In these cases, the signal becomes indistinguishable from shot noise, and optimization cannot proceed. As a result, the search for architectures that provably avoid barren plateaus has become a central focus in the design of trainable quantum models.

In parallel, advances in classical simulation techniques had direct consequences for quantum machine learning. Significant progress was made in *classical simulation* of quantum circuits, that is, computing their outputs using classical algorithms without needing access to a quantum device. In general, this is a hard problem: the resources needed for a classical computer to naively track the full quantum state grow exponentially with the number of qubits. However, some quantum circuits are easy to simulate. This is often the case when the circuit is shallow (i.e., has few gates), or when it is restricted to specific gate sets, such as the Clifford group [12]. Efficient simulation is also possible when circuits have special algebraic structure [13], or when they are highly noisy [14–16], making either their average output easy to simulate or their output distribution effectively easy to sample from. Understanding which circuits are classically simulable is essential because these circuits cannot provide a quantum advantage by nature.

In particular, many of the PQC architectures that were specifically designed to avoid barren plateaus were later shown to be classically simulable [13, 17]. This exposed a growing tension: circuits that are easy to train are often also easy to simulate classically, and therefore unlikely to provide any true quantum advantage [18]. This narrowing gap between trainability and simulability has forced the community to critically reassess the role of PQCs in quantum machine learning. While the early enthusiasm has been tempered by these findings, the search for quantum learning advantages continues, looking for learning algorithms that not merely *use* quantum computers but rather *need* quantum computers.

Despite these setbacks, there is rigorous evidence that quantum computers can outperform classical ones on certain learning tasks. These results are known as *learning separations*: formal examples where a quantum learner can efficiently learn a function, while any classical learner would require super-polynomially more resources. The first such separations were based on cryptography-inspired tasks such as the factoring problem for Blum integers [19] and the discrete logarithm problem [20]. The latter is a central problem in cryptography and is at the heart of most protocols securing communication today. More recent works have uncovered learning separations that go beyond cryptographic tasks [21, 22], showing that quantum advantage can emerge when the target function relates to the set of hardest problems quantum computers are expected to solve efficiently. These are valuable because they show that a quantum advantage in learning is possible in principle, but this comes with a caveat: how these problems may relate to physically relevant tasks remains an open question. It is a challenge to find quantum learning separations that have an impact on machine learning applications we care about for real-world datasets.

This thesis has the overarching goal of looking for quantum advantages. While this theme is broad, to contribute to its elucidation we will focus on two main aspects. On the one hand, deepening our understanding of the capacities of quantum machine learning models, exploring the universality of PQC-based models, and finding provable quantum learning advantages. And on the other hand, defining and analyzing the computational complexity of simulating classical and quantum systems with bosonic systems. We detail these two aspects in the following section, which also outlines the research questions that guide this thesis.

1.2. Research questions

In order to make the research questions more precise, we briefly clarify some basic terminology, which we will make fully precise later in Chapter 2.

First, we unpack the theoretical capabilities of parameterized quantum circuits (PQCs) as models for two core tasks in learning: supervised prediction and generative sampling. In *supervised learning*, the model is trained on labeled example pairs to learn an underlying mapping. The learning algorithm should provide a model that predicts outputs close to the correct ones for unseen inputs. By contrast, *generative modeling* is given unlabeled samples and aims to learn the distribution of the data itself. The learning algorithm should provide a model that generates new samples from a distribution that is as close as possible to the target distribution.

Expressivity is a fundamental concept across both supervised and generative learning: it quantifies a model's ability to represent all functions or probability distributions. Formally, given a target space and a model class, we say a model is more expressive if it can approximate a larger portion of the target space. The highest level of expressivity is called *universality*, where the model class is dense in the target space: any target element can be approximated up to arbitrary precision by some model element. This property is well-known in classical settings. For example, neural networks satisfy universal approximation theorems under mild conditions [23, 24].

This opens the question of whether PQCs can achieve universality, and if so, under what conditions. Some universality results have been established for PQCs, showing that they can approximate any continuous function [25, 26]. In the context of both supervised learning and generative learning, we sharpen this question in the case of limited resources, and this leads us to the first research question.

Research Question 1
What are the minimal resources needed to achieve universality with parameterized quantum circuits?

Secondly, we turn our attention to the question of when quantum computers can provide a provable learning advantage over classical ones. As mentioned in the preface, most variational methods are heuristic and lack theoretical guarantees. Yet, there are notable exceptions where quantum learning advantage can be proven rigorously. These proofs rely on statistical learning theory, specifically the *probably approximately correct* (PAC) learning framework [27, 28]. The PAC framework provides a formal definition for a learning algorithm to be successful, by upper-bounding the probability that the algorithm exceeds an error threshold on unseen data. If the resources in terms of computation time and size of dataset needed for the algorithm to be successful are polynomial in the size of the training set, we say that the class is efficiently PAC-learnable. This framework gives us a rigorous benchmark to claim that a quantum learning algorithm is efficient, moving us beyond heuristics.

If, in addition to being efficient, the quantum learning algorithm can learn a function that is provably hard for any classical algorithm to learn, we say that we have a *learning separation*. As mentioned in the preface, several learning separations were proven, first on cryptographic tasks [19, 20] and later for tasks associated with PromiseBQP-complete problems [21, 22]. While these previous works did not focus on PQCs, in this thesis, we investigate PQC-based methods that allow a learning separation, leading us to the second research question.

Research Question 2
How can we leverage our knowledge of parameterized quantum circuits to construct practical provable quantum learning advantages?

So far, we have focused our attention on the advantages provided by qubit-based quantum computers; but *continuous-variable quantum computing* (CVQC) constitute an alternative computational model. CVQC leverages quantum systems with continuous degrees of freedom, such as the position and momentum of particles, to perform computation. Alternatively, such states can be understood as vectors in an infinite-dimensional space, called the Fock space, which is in contrast to the discrete qubit-based model, where quantum information is encoded in two-level systems (qubits). This paradigm typically arises from the physics of bosonic systems, where quantum information is encoded in the states of harmonic oscillators, such as modes of light or vibrational modes in ions. Within CVQC, a prominent role is played by *Gaussian states*, whose quadrature operators measurements can be described as Gaussian distributions. Operations called Gaussian gates (e.g., displacements, rotations, and squeezers) preserve the Gaussian property of states. Gaussian states evolved under Gaussian gates on n modes are classically simulable. However, adding non-Gaussian gates lifts this restriction and enables universal quantum computation in the continuous-variable regime [29]. The complexity of unrestricted bosonic systems has been studied in [30] and related to complexity classes such as EXP. In the final part of this thesis, we turn to bosonic systems with some added restrictions and relate their computational complexity to that of classical systems and qubit-based quantum computers

In Chapter 2, we introduce the complexity classes mentioned in this paragraph at greater length, but we provide a high-level introduction here first. We briefly mentioned in the preface the set of hardest problems that quantum computers are expected to solve efficiently. These are known as *PromiseBQP-complete* problems, that is the hardest problems that are in BQP (Bounded-error Quantum Polynomial time). BQP [31] is the complexity class of all decision problems that can be decided by a quantum computer in polynomial time with bounded error. If any of the BQP-complete problems was to be solved efficiently on a classical computer, this would violate widely believed complexity theory assumptions. Beyond BQP, there are even harder quantum complexity classes, such as QMA (Quantum Merlin-Arthur), which is the quantum analog of NP [32, 33]. More powerful still is PostBQP, which allows post-selection on specific measurement outcomes—conditioning on events with exponentially small probability [34]. Remarkably, PostBQP includes QMA and coincides with the classical complexity class PP [34], situating it as a powerful tool to probe the limits of quantum computation.

This leads us to the following research question, which explores how the complexity of bosonic systems maps onto the complexity landscape of classical and qubit-based systems.

Research Question 3

How does the computational complexity of bosonic systems relate to that of classical systems and qubit-based systems?

1.3. List of contributions

The following works form the basis of this dissertation.

[35] A. Barthe and A. Pérez-Salinas. Gradients and frequency profiles of quantum re-uploading models. Quantum, vol. 8, p. 1523 (2024).

[36] A. Barthe, M. Grossi, S. Vallecorsa, J. Tura, and V. Dunjko. Parameterized quantum circuits as universal generative models for continuous multivariate distributions. npj Quantum Inf, vol. 11, no. 1, p. 121 (2025)

[37] A. Barthe, M. yaghubi, M. Grossi, and V. Dunjko. Quantum Advantage in Learning Quantum Dynamics. arXiv:2506.17089 (2025).

[38] A. Barthe, M. Grossi, J. Tura, and V. Dunjko. Continuous variables quantum algorithm for solving ordinary differential equations. IEEE International Conference on Quantum Computing and Engineering (2023)

[39] A. Barthe, M. Cerezo, A. T. Sornborger, M. Larocca, and D. García-Martín. Gate-Based Quantum Simulation of Gaussian Bosonic Circuits on Exponentially Many Modes. Phys. Rev. Lett., vol. 134, no. 7, p. 070604 (2025).

The following publications were co-authored during the course of the PhD and are not included in this thesis.

[40] M. Sahebi, A. Barthe, Y. Suzuki, Z. Holmes, and M. Grossi. On Dequantization of Supervised Quantum Machine Learning via Random Fourier Features. arXiv: arXiv:2505.15902 (2025).

[41] A. Pérez-Salinas, M. Y. Rad, A. Barthe, and V. Dunjko. Universal approximation of continuous functions with minimal quantum circuits. arXiv: arXiv:2411.19152 (2025).

1.4. Overview of this thesis

Chapter 3 is based on the peer-reviewed journal article [35], the first on the list in the previous section. In this chapter, we analyze the average behavior of a class of PQCs called quantum re-uploading models. These models iteratively encode data into quantum circuits, with trainable gates in between. We focus on theoretical aspects of QRU focusing on their trainability and their expressivity in a supervised learning context. We study the gradients of cost functions associated with these models and quantify the influence of data encoding layers on their trainability. Furthermore, we explore the expressivity of quantum re-uploading models by examining properties of the Fourier decomposition of the models. Our results demonstrate that, on average, this decomposition follows a Gaussian profile and that high-frequency components vanish.

Chapter 4 is based on the peer-reviewed journal article [36], the second on the list in the previous section. In this chapter, we move to a generative modeling context for multidimensional distributions of continuous random variables. We formalize a class of quantum generative models for such distributions, which we call *expectation value samplers.* The idea is to use a PQC to sample from a distribution defined by the expectation values of a set of observables, which can be efficiently computed using quantum circuits. In this chapter, we derive tight necessary and sufficient conditions on the quantum resources required to achieve universality for such models. We do so by leveraging existing knowledge on parameterized quantum circuits, demonstrating how these models can represent complex distributions. Furthermore, we identify a physically relevant distribution that these models can efficiently represent.

Chapter 5 is based on the preprint [37], the third on the list in the previous section. Building on the refined understanding of PQCs and their connection to Fourier analysis, we investigate how quantum feature maps can be constructed to achieve provable learning separations. Feature maps are a key component of many machine learning algorithms, transforming the input data into a higher-dimensional space where it is easier to learn patterns. In this chapter, we introduce an algorithm for extracting the Fourier spectrum of PQC-based functions, enabling the design of feature maps that are likely intractable for classical computers. Leveraging this approach, we demonstrate a quantum learning advantage for a specific learning task based on PQC-based functions. Furthermore, we extend this framework to derive a learning separation for a physically relevant task involving Hamiltonian time evolution. These results highlight the potential of PQCs in constructing quantum learning advantages for tasks grounded in quantum dynamics, while emphasizing the necessity of fault-tolerant quantum computation for such applications.

Chapter 6 is based on the peer-reviewed proceedings paper [38], the fourth on the list in the previous section. We propose a method for using continuous variable quantum computers to solve nonlinear differential equations, leveraging the Koopman von Neumann framework. By working directly in the continuous variable quantum computing paradigm, we avoid the truncation errors associated with finite-dimensional approximations. Our algorithm compiles a sequence of Gaussian and non-Gaussian gates to approximate the time evolution of the Koopman von Neumann Hamiltonian for polynomial ordinary differential equations. This method is particularly suited for solving initial distribution problems, where the evolution of probability distributions is studied rather than individual initial conditions.

Chapter 7 is based on the peer-reviewed journal article [39], the fifth on the list in the previous section, with additional unpublished results on larger complexity classes. Quantum computers are known to provide advantages for simulating certain large classical dynamics. Inspired by quantum algorithms for exponentially many coupled oscillators [42], we develop a quantum simulation framework for bosonic Gaussian states on exponentially many modes. This framework encodes the first and second moments of quadrature operators into qubit states, enabling efficient simulation of Gaussian bosonic circuits using gate-based quantum computers. We introduce a mapping between bosonic gates and qubit gates, allowing particle-preserving Gaussian bosonic circuits to be simulated efficiently. Furthermore, we identify a PromiseBQP-complete problem within this framework, demonstrating that simulating Gaussian bosonic circuits on exponentially many modes is as powerful as universal quantum computation. Finally, we show that adding a layer of squeezing gates to interferometers boosts the complexity of the problem to PostBQP-complete.

CHAPTER 2

Background

2.1. Statistical Learning Theory

2.1.1. Concept classes and PAC learning

Defining success for a computational task, such as factoring a number or solving an optimization problem, is relatively easy. However, defining success for learning tasks is more subtle. In this section, we introduce the *Probably Approximately Correct (PAC)* learning framework [27], which is a formal mathematical framework for learning tasks, which we will heavily rely on in Chapter 5.

PAC learning theory provides a formal mathematical framework for learning tasks, it revolves around so-called concept classes $\mathcal{C}$.

Definition 2.1 (Concept class)
A concept class is a set of functions called concepts c_α*.*

$$\mathcal{C} = \{c_\alpha : x \in \mathcal{X} \to y = c_\alpha(x) \in \mathcal{Y}\}_{\alpha \in \mathcal{A}} \,. \tag{2.1}$$

The goal in PAC learning is to approximate an unknown concept from a specified class given access to a dataset of examples labeled by the concept. That is, for a fixed unknown concept c_{α_0} in the concept class, we are given T input-output pairs. The inputs x_t are sampled from a probability distribution D over the input domain $\mathcal{X}$, and the outputs are $c_{\alpha_0}(x_t)$. This dataset is denoted by $\mathcal{S}$.

$$\mathcal{S} \coloneqq \{(x_t, y_t = c_{\alpha_0}(x_t))\}_{t\in[1,T]},\ x_t \overset{\text{i.i.d.}}{\sim} D\,. \tag{2.2}$$

Using such a dataset, the goal of a learning algorithm is to output a *hypothesis* function $h : \mathcal{X} \to \mathcal{Y}$ from a set of possible hypotheses $\mathcal{H}$, called the hypothesis class. This hypothesis function is used to assign labels to unseen data x_{new}. When the learning is successful, the assigned labels $\hat{y}_{\text{new}} = h(x_{\text{new}})$ are ε-close to the true label with high probability $1-\delta$. A pair $(\mathcal{C}, D)$ is PAC learnable if there exists an algorithm that produces a hypothesis h given only polynomially many examples, that is T scales at most polynomially with δ^{-1}, ε^{-1} and the size of the input. Adding to PAC learnability a notion of computational costs, a concept class $\mathcal{C} = \{\mathcal{C}_n\}_n$[1] increasing in size n, which yields the concept of *efficient* PAC learning, defined as follows.

Definition 2.2 (Efficient PAC learnability)
The concept class $\mathcal{C} = \{\mathcal{C}_n\}_n$ *is efficiently PAC learnable if for all* $\varepsilon \geq 0$*, all* $0 \leq \delta \leq 1$*, all* $n \in \mathbb{N}$ *there exists a* $\text{poly}(\varepsilon^{-1}, \delta^{-1}, n)$*-time algorithm* $\mathcal{A}$ *such that for any target concept* $c : \mathcal{X}_n \to \mathcal{Y}_n$ *in* $\mathcal{C}_n$ *and any target distribution* $\mathcal{D}_n$ *on* $\mathcal{X}_n$*, if* $\mathcal{A}$ *receives in input a training dataset* $\{(x_t, c(x_t))\}_{t\in[0,T]}$ *of* $\text{poly}(\varepsilon^{-1}, \delta^{-1}, n)$*-size* T*, then with probability at least* $1-\delta$ *over the random datasets, the learning algorithm* $\mathcal{A}$ *outputs a specification of a hypothesis function* $h = \mathcal{A}(T, \varepsilon, \delta)$ *running in* $\text{poly}(\varepsilon^{-1}, \delta^{-1}, n)$*-time that satisfies*

$$\text{Pr}_{x\sim\mathcal{D}_n}(|h(x) - c(x)| \leq \varepsilon) \geq 1 - \delta. \tag{2.3}$$

The average error of the hypothesis h *is also referred to as the* true risk.

We say that a concept class is *classically efficiently learnable*, if it is efficient PAC learnable with both $\mathcal{A}$ and h running in polynomial time on a classical computer. Conversely, we say that a concept class is *quantum efficiently learnable*, if it is efficient PAC learnable with both $\mathcal{A}$ or h running in polynomial time on either a quantum or a classical computer. In Chapter 5, we will prove that a concept class based on quantum dynamics is quantum efficiently learnable, while it is not classically efficiently learnable.

---

[1] Here the concept class is divided into sub-classes specific to input size n to make the dependence more explicit.

2.1.2. Covering numbers

In this section, we introduce the concept of covering numbers, which is a measure of the complexity or the size of an infinite set of functions, and play a crucial role in PAC learning theory.

Definition 2.3 (Covering number)
The covering number of a set of functions $\mathcal{C}$ with respect to a distance d is the smallest number of balls of radius ε needed to cover the set $\mathcal{C}$, denoted as $N(\mathcal{C}, d, \varepsilon)$.

$$N(\mathcal{C}, d, \varepsilon) = \min_{\mathcal{H}} |\mathcal{H}| \textit{ such that } \forall f \in \mathcal{C}, \exists h \in \mathcal{H} : d(f, h) < \varepsilon\,. \tag{2.4}$$

The logarithm of the covering number of a concept class can be used to analyze generalization performance, as it appears in upper bounds on the number of samples needed to learn a concept class. We provide bounds on the covering numbers of the proposed concept classes in Chapter 5.

2.2. Quantum Computing

This section introduces the minimal fundamentals of qubit-based quantum computing needed for the rest of this thesis. We cover quantum states, quantum gates, and measurements as the three pillars of quantum computation. We refer the reader to [43] for a more complete introduction to quantum computing.

2.2.1. Quantum states

The fundamental unit of quantum information is the *qubit*, a two-level quantum system represented by a vector in a two-dimensional Hilbert space $\mathbb{C}^2$. A qubit's state is a unit vector $|\psi\rangle \in \mathbb{C}^2$:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \quad \alpha, \beta \in \mathbb{C},\ |\alpha|^2 + |\beta|^2 = 1\,. \tag{2.5}$$

Here, $|0\rangle$ and $|1\rangle$ form the so-called computational basis. All quantum computations ultimately refer to measurement outcomes in this basis, while the global phase of a state is unobservable.

For systems of n qubits, the joint state lives in a 2^n-dimensional Hilbert space formed by tensor products of single-qubit spaces. $\otimes$ denotes the tensor product. This operation defines the structure of multi-qubit systems as follows,

$$\otimes : \mathbb{C}^n \times \mathbb{C}^m \to \mathbb{C}^{mn}\,. \tag{2.6}$$

For example, if $|\psi_1\rangle$ and $|\psi_2\rangle$ are single-qubit states, the joint state of the two-qubit system is given by

$$\begin{aligned}|\psi_1\rangle \otimes |\psi_2\rangle &= (\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle) \\ &= \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle\,, \end{aligned} \tag{2.7}$$

In general, a full n-qubit system can be described as

$$|\psi\rangle = \sum_{i=0}^{2^n-1} c_i|i\rangle\,, \quad \sum_i |c_i|^2 = 1\,, \tag{2.8}$$

where $|i\rangle$ runs over all the n-bitstrings, each corresponding to a computational basis state $|b_1b_2\cdots b_n\rangle$ with $b_j \in \{0,1\}$.

While many quantum algorithms assume *pure states*, noise and decoherence often require the more general framework of *mixed states*, described by a density matrix ρ. Mixed states form a classical mixture of pure states $|\psi_k\rangle$ according to a probability distribution $\{p_k\}$. ρ can be expressed as the linear combination of their projectors $|\psi_k\rangle\langle\psi_k|$, as follows,

$$\rho = \sum_k p_k|\psi_k\rangle\langle\psi_k|\,, \quad \mathrm{Tr}(\rho) = 1,\ \rho \geq 0\,. \tag{2.9}$$

2.2.2. Quantum gates

Quantum gates are unitary operators U acting on the Hilbert space. They satisfy

$$U^\dagger U = UU^\dagger = I, \tag{2.10}$$

and evolve quantum states according to

$$|\psi\rangle \mapsto U|\psi\rangle\,. \tag{2.11}$$

The most commonly used quantum gates are listed in Table 2.1, along with their matrix representations.

A gate set is *universal* if it can approximate any unitary U on n qubits to arbitrary precision. Clifford gates alone, generated by H, S, and CNOT, are not universal and are classically simulable [12]. Adding a non-Clifford gate, such as T, yields a universal gate set.

The time evolution of a closed quantum system is governed by the

Table 2.1.: Common single-qubit and two-qubits quantum gates

Gate	Symbol	Matrix Representation
Pauli-X	X	$\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$
Pauli-Y	Y	$\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}$
Pauli-Z	Z	$\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$
Hadamard	H	$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$
Phase	S	$\begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}$
T gate	T	$\begin{pmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{pmatrix}$
CNOT	CX	$\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}$

Schrödinger equation,

$$i\frac{d}{dt}\left|\psi(t)\right\rangle = H\left|\psi(t)\right\rangle , \tag{2.12}$$

where H is the system Hamiltonian (a Hermitian operator). The time evolution corresponds to a unitary evolution operator as follows,

$$U(t) = e^{iHt} . \tag{2.13}$$

Parameterized quantum gates arise naturally when simulating Hamiltonian evolution for fixed time. These take the form $R_\alpha(\theta) = e^{i\theta H/2}$ where H is a Hermitian operator called the generator. These are most commonly some Pauli operators, such as:

$$R_x(\theta) = e^{i\theta X/2}, \tag{2.14}$$

$$R_z(\theta) = e^{i\theta Z/2}. \tag{2.15}$$

Such gates are called *parameterized quantum gates* and are central to variational quantum circuits.

2.2.3. Measurements

Measurements extract classical information from quantum states. In the computational basis, the probability $p(i)$ of observing outcome $|i\rangle$ is given by the Born rule, that is, the norm of the projection of the state onto the computational basis state $|i\rangle$. This can be expressed with a dot product $\langle\cdot|\cdot\rangle$ as follows,

$$p(i) = |\langle i|\psi\rangle|^2. \tag{2.16}$$

More generally, measurements are described by a set of operators $\{M_i\}$ forming a *positive operator-valued measure* (POVM) [43], satisfying:

$$\sum_i M_i^\dagger M_i = I. \tag{2.17}$$

If outcome i is observed, the post-measurement state is

$$|\psi'\rangle = \frac{M_i\,|\psi\rangle}{\sqrt{p(i)}}, \quad p(i) = \langle\psi|\, M_i^\dagger M_i\, |\psi\rangle\,. \tag{2.18}$$

In quantum algorithms, a frequent task is estimating the expectation value of an observable O. An observable is a Hermitian operator, and therefore has real eigenvalues $\{\lambda_k\}_k$ and its eigenvectors $\{|\lambda_k\rangle\}_k$ form an orthogonal basis and are the possible measurement outcomes. An observable O can be expressed as follows,

$$O = \sum_k \lambda_k\, |\lambda_k\rangle\!\langle\lambda_k|\ . \tag{2.19}$$

The expectation value is given by

$$\langle O\rangle = \langle\psi|\, O\, |\psi\rangle = \sum_k \lambda_k |\langle\lambda_k|\psi\rangle|^2\,. \tag{2.20}$$

This value may be estimated by running the circuit multiple times (shots) and computing the empirical average of the eigenvalues corresponding to the measurement outcomes. The uncertainty of this estimate scales as $\sigma/\sqrt{N}$ where σ^2 is related to the spectrum of O and N is the number of shots [5].

2.3. Parameterized Quantum Circuits

In this section, we provide more details on concepts related to concepts related to the expressivity, trainability, and hardness of Parameterized Quantum Circuits (PQC), which are the main focus of the first part of this thesis.

2.3.1. Expressivity

Expressivity in machine learning and computer science characterizes the ability of a model class $\mathcal{H}$ to represent functions within a target space $\mathcal{C}$. Formally, $\mathcal{H}$ is considered more expressive relative to $\mathcal{C}$ when it can approximate more elements of $\mathcal{C}$. The highest level of expressivity, universality, occurs when $\mathcal{H}$ is dense in $\mathcal{C}$, meaning any target in $\mathcal{C}$ can be approximated to arbitrary precision by some element in $\mathcal{H}$. Universality has been proven for several classes of classical machine learning models, including neural networks with the Universal Approximation Theorem [23, 24].

Parameterized Quantum Circuits (PQCs) form the computational foundation for diverse quantum models and variational algorithms [4], central to hybrid quantum-classical approaches in the current noisy intermediate-scale quantum (NISQ) era. Here, we formalize their structure, applications, and expressivity across diverse tasks. First, we highlight that depending on the context of their usage, the expressivity of Parameterized Quantum Circuits does not always have the same meaning. More precisely, expressivity is relative to the nature of the target space $\mathcal{C}$, which can be, for some, the set of all quantum states, while for others it is a functional space.

We distinguish between two primary uses of parameterized quantum circuits (PQCs): input-free (used as ansatzes for variational problems) and data-dependent (used in learning tasks). This distinction is fundamental to understand Chapter 3, as we transfer results for input-free Parameterized Quantum Circuits to data-dependent ones. We use universality results in a supervised context to transfer them to a generative context in Chapter 4. Finally, we use the connection between Parameterized Quantum Circuits and Fourier analysis in Chapter 5 to design a quantum learning algorithm.

Input-free Parameterized Quantum Circuits

A Parameterized Quantum Circuit is a sequence of fixed and parameterized quantum gates applied to an initial state (typically $|0\rangle^{\otimes n}$). We focus on

PQC such that their unitary evolution is defined as:

$$|\psi(\theta)\rangle = U(\theta)\,|0\rangle = \prod_{j=1}^{M} W_j e^{iV_j\theta_{m_j}}\,, \tag{2.21}$$

where $\{V_j\}_j$ are Hermitian generators (e.g. Pauli Operators), $\{W_j\}_j$ are fixed unitaries (e.g. entangling gates), and $\theta \in \mathbb{R}^M$ are trainable real parameters. The output is often the expectation value of an observable O:

$$h(\theta) = \langle 0|\, U^\dagger(\theta) O U(\theta)\,|0\rangle\,. \tag{2.22}$$

Such parameterized circuits can be used in a variety of contexts. In *Variational Quantum Eigensolver* [7], O is chosen as the Hamiltonian we seek to find the ground energy of. This is done by finding the parameters of the state $|\psi(\theta)\rangle$, also called an ansatz minimizing the function $h(\theta)$. Similarly, PQCs may also be used to solve optimization problems, with *Quantum Approximate Optimization Algorithm* [8], where the function to minimize is mapped to an observable O. PQC are also used in a generative modeling context, with *Quantum Circuit Born Machines* [44, 45] (QCBM), where they are used to sample bitstrings based on the Born rule. The goal is to find a set of parameters θ such that sampling bitstrings from the state $|\psi(\theta)\rangle$ yields a distribution over n-bitstrings as close as possible to a target distribution. In all of these cases, the expressivity of the PQC is relative to the Hilbert space. The closer the set $\{|\psi(\theta)\rangle\}_{\theta\in\mathbb{R}^M}$ is to the set of all possible quantum states, the more expressive the model is considered (to the exception that local phases do not matter for QCBM).

Finally, for *Variational Quantum Compilation*, given a target unitary V the goal is to optimize the parameters θ such that the parameterized quantum circuit $U(\theta)$ approximates V. This may be used to compile shallower circuits when the target unitaries, such as time evolution [46], or also approximating a diagonalization such as in variational fast forwarding [47]. In that case, the expressivity is relative to the set of all n-qubits unitaries and sometimes is measured relative to unitary t-designs [48].

Parameterized Quantum Circuits with inputs

Parameterized quantum circuits can, in general, have both *inputs* and *parameters*. So far, we considered PQC with only parameters, serving as ansatzes for optimization problems rather than machine learning in the sense of learning from data. In this subsection, we consider PQC with inputs, which are used to upload data from machine learning tasks. In this thesis, we consider two different machine learning contexts, supervised learning and generative modeling. In supervised learning, given pairs of input-output $(x_t, y_t)_t$, the goal is to return a function h that mimics the underlying relationship f between the input and the output $y = f(x)$. In generative modeling, given a set of inputs sampled from an unknown distribution p, the goal is to return a sampling algorithm whose underlying distribution q approximates the target distribution.

It is possible to make a quantum state that is input-dependent, or, in other words, onto which data is encoded. We discuss different architectures of input-dependent Parameterized Quantum Circuits in the case where inputs are D-dimensional real vectors $x \in \mathbb{R}^D$, as follows,

$$|\psi(x,\theta)\rangle = U(x,\theta)\,|0\rangle \ . \tag{2.23}$$

We do not cover all architectures exhaustively, but we cover some of the most commonly used ones. A first natural encoding is the so-called amplitude encoding, where the overlap of the state with the k-th computational basis state is the k-th coordinate of the normalized data as follows, resulting in a qubit-dense encoding, where $n = \log(D)$ qubits are needed.

$$U(x)\,|0\rangle = \|x\|_2^{-1/2} \sum_{d=0}^{D-1} x_d\,|d\rangle \ . \tag{2.24}$$

Another option to encode data onto a quantum state yields product states, and requires significantly more qubits with $n = D$.

$$U(x)\,|0\rangle = \otimes_k R_y(x_k)\,|0\rangle = \otimes_k(\cos(x_k)\,|0\rangle + \sin(x_k)\,|1\rangle)\,. \tag{2.25}$$

However, such models have limited expressivity with respect to the set of all possible functions. So-called *quantum reuploading models* were introduced by [25, 49] to propose a richer expressivity. Such circuits interleave data-dependent gates and parameterized gates. The most common form for such models is

$$|\psi(x,\theta)\rangle = U(x,\theta)\,|0\rangle = \prod_{j=1}^{L} W_j(\theta) e^{iV_j x_{m_j}} \ , \tag{2.26}$$

where each $W_j(\theta)$ is a parameterized quantum circuit and $\{V_j\}_j$ are Hermitian generators. The terms of this product are often called layers. Other forms of quantum reuploading models with alternative data encoding gates are considered in e.g. [41].

Usually, the expectation value of an observable O is returned, and the parameterized quantum circuit is the key component of a parameterized function, as follows

$$h_\theta : x \to \langle 0| U^\dagger(x,\theta) O U(x,\theta) |0\rangle \ . \tag{2.27}$$

Typically, this parameterized set of functions is used in supervised learning, where the goal is to approximate the relation between inputs and labels. That is, for a dataset is $\{(x_t, y_t = f(x_t))\}$ for an unknown function f we optimize the parameters θ such that h_θ approximate f.

The expressivity of such circuits is well-understood, and their universality has been proven for several architectures [25, 26, 41]. In general the function they represent can be expressed as generic trigonometric polynomials [26] (GTP), that is there exist a finite set of d-dimensional real vectors $\Omega = \{\omega_l \in \mathbb{R}^d\}_l$ that are usually called frequencies, and a vector of complex numbers $x \in \mathbb{C}^{|\Omega|}$ such that

$$h_\theta(x) = \sum c_l(\theta) e^{i\omega_l \cdot x} \ . \tag{2.28}$$

Leaving the generators V_j free, with, for example, the possibility to have scaling parameters (e.g. $V_j = \alpha Z$), yields universality on a single qubit [25].

A version of this model restricted to Pauli encodings, that is, all V_j must be Pauli strings, is often used, as in the definition below.

Definition 2.4 (Pauli encoding)
A Pauli-encoded circuit is a parameterized quantum circuit on n *qubits* $U : x \in [0,1]^D \to \mathcal{U}(2^n)$. *It is composed of* $N_f \in \mathrm{poly}(n)$ *fixed unitary gates and* $L \in \mathrm{poly}(n)$ *parameterized gates* $\{V_l(x) := e^{i\pi P_l x_{i_l}}\}_{1\leq l \leq L}$ *where* P_l *are Pauli strings.*

When generators are restricted to the set of Pauli strings, the set of frequencies is restricted to a finite set of integers $\Omega = [-L, +L]^d$, and the GTP is de facto a finite Fourier decomposition and a universal model [26].

Measuring the expectation value of some observable O for such Pauli encoded circuits results in what we call a *PQC-based function*, an input-output mapping as follows,

$$f(x) = \langle 0| U^\dagger(x) O U(x) |0\rangle \ . \tag{2.29}$$

In the case of quantum reuploading models with Pauli encodings as in

Definition 2.4, we have that

$$f(x) = \sum_{l \in [-2L, 2L]^D} b_l e^{i\pi x \cdot l} \, . \tag{2.30}$$

We call the coefficients b_l the *Fourier coefficients* of the *PQC-based function* f.

So far, we have seen ways to use parameterized quantum circuits as explicit models, where the labels are specified by the output of the circuit, but it is also possible to use them as implicit models, such as in kernel methods. In this other option, the function $x \to |\psi(x)\rangle$ becomes a feature map, and a quantum computer is used to compute the corresponding kernel values as $k(x, x') = |\langle \psi(x) \mid \psi(x') \rangle|^2$. The difference between three different models, linear models (where $U(x, \theta) = U'(x)U''(\theta)$), Quantum Reuploading models, and quantum kernels have been studied in [50]. This work also shows equivalences between these models at the cost of more computational resources.

It is crucial to note that the notion of expressivity is very different between input-free PQC and data-dependent PQC. In the first case, the expressivity is relative to the Hilbert space, while in the data-dependent it is relative to a set of functions for supervised learning. We explore this distinction deeper in Chapter 3. For generative modeling, the expressivity is relative to the set of multivariate distributions, we explore this further in Chapter 4 and establish universality results. Finally, we use the connection between PQC and Fourier analysis to prove quantum advantage in learning quantum dynamics in Chapter 5.

2.3.2. Trainability

The training of parameterized quantum circuits (PQCs) involves an optimization procedure that searches for parameter sets minimizing a cost function $\mathcal{L}(\theta)$. The feasibility of this optimization task is captured by the concept of *trainability*, which has been extensively studied in quantum machine learning contexts [4, 11].

In typical implementations, this is done in a hybrid architecture with calculation taking place alternatively in quantum and classical computers. A quantum computer is used to estimate $\mathcal{L}(\theta)$ and its gradients $\partial_\theta \mathcal{L}(\theta)$. A classical computer then updates the parameters using gradient descent, iterating until convergence.

In quantum machine learning, trainability challenges emerge from characteristics of the cost landscape, including non-convex landscapes with numerous local minima [51]. A fundamental limitation is the *barren plateau (BP) phenomenon* [11], where gradients vanish exponentially with the system size

$$\mathbb{E}_\theta[\partial_\theta \mathcal{L}(\theta)] \in \mathcal{O}(e^{-n}) \tag{2.31}$$

Other formulations were found to be equivalent, such as the variance of the cost function itself vanishing exponentially.

As the estimation of gradients is affected by shot noise, when they are exponentially small, they provide no useful information for parameter updates, and this leads to a situation where the optimization becomes ineffective. This phenomenon is prevalent in many PQC architectures, particularly those with high expressivity [52], entanglement [53], global observables [54], and noise in quantum hardware [55]. Gradient-free optimization methods (e.g., COBYLA, SPSA) also fail in BP regimes due to exponentially small cost function variance [56]. Therefore, understanding how the gradients' magnitude scales with the number of qubits is crucial; we address ways to bound them in Chapter 3.

To address the BP problem, several architectures have been proposed that are provably BP-free, meaning they can be trained efficiently without encountering barren plateaus. Among others, there are *Symmetry-preserving ansatzes* [57], *Shallow circuits* with $\mathcal{O}(\log n)$ depth with local observables [54] and PQCs with polynomially sized *Dynamical Lie Algebras* [58]. Initialization strategies have also been proposed to mitigate the effects of BP in PQCs, a strategy also called warm start [59].

However, these BP-resistant architectures were matched with classical strategies to simulate them efficiently. In Section 2.3.3 we provide examples of such classical algorithms that are able to replace PQCs in some cases. We review existing results about the classical hardness, or lack thereof, of several learning tasks.

2.3.3. Hardness of learning

Trainability vs simulability

Recent theoretical work [18] has established a connection between barren plateau (BP) mitigation strategies and classical simulability in parameterized quantum circuits. It presents evidence that architectures with provable BP absence are classically simulable with polynomial-time classical algorithms. This follows from the observation that BP avoidance strategies inherently confine computations to polynomially-sized subspaces of the full Hilbert space, which can be efficiently characterized and simulated classically. This pattern holds across several major BP-mitigation strategies including shallow circuits [54], geometrically local observables [60], symmetry-preserving ansatzes [57], and tensor-network based designs [61, 62]. This work highlighted the thin gap between trainability and simulability, and pushed the community to find alternative approaches to quantum machine learning.

Random Fourier Features

The concept of finding efficient classical alternatives to quantum algorithms is known as dequantization. In addition to the aforementioned results on trainability and simulability, another classical learning algorithm was shown to be able to dequantize quantum machine learning in some cases. The central idea of this dequantization method we present now revolves around the fact that most used PQC architectures can be expressed as generic trigonometric polynomials (GTP). This makes them a natural candidate for classical learning algorithms based on so-called Random Fourier Features. The idea is to move away from trying to simulate PQCs, but rather to mimic the associated hypothesis class. Random Fourier Features are defined for a probability distribution over a set of frequencies $p_\omega : \Omega \to [0,1]$. This yields a random feature map Φ, as a stack of L trigonometric functions for frequencies sampled from the distribution p_ω, as follows,

$$x \xrightarrow{\Phi} [e^{i\omega_l \cdot x}]_{l\in[1,L]} \,, \omega_l \overset{i.i.d}{\sim} p_\omega \,. \tag{2.32}$$

A number of sufficient conditions were found for RFF-based algorithms to have a performance matching or exceeding that of PQC-based algorithms in terms of true risk (definition in Section 2.1). Several theorems [40, 63] have been found to cover most cases: explicit PQC based expression as well as implicit models based on kernels, for classification and regression tasks. Overall, the sufficient conditions are properties of the Fourier transform of the optimal PQC function, normalized to be a distribution q_ω, and the distribution of the RFF p_ω.

Theorem 2.1 (RFF dequantization theorem, informal)
For any learning task, the performance in terms of true risk of an RFF-based algorithm with a probability distribution p_ω exceeds or matches that of any PQC machine learning scheme outputting the optimal hypothesis function, for which we write the associated frequency probability q_ω, if all the following conditions are met:

1. *p_ω is easy to sample from,*

2. *q_ω and p_ω are aligned, that is, their dot product is at least inverse polynomial,*

3. *p_ω is concentrated, that is, its max norm is at least inverse polynomial.*

Dequantization is usually defined as the existence of a p_ω that satisfies such conditions. But then the question of how easily one may find such a distribution remains open [63]. Leaving the domain of provable performance, this algorithm can still be a good heuristic in its own right.

Provable learning separations

In the light of these dequantization results, it is crucial to identify regimes where quantum computers are not merely *used* for machine learning tasks, but truly *necessary.* These situations are called *learning separations* and are characterized by the existence of learning tasks that classical learners cannot address efficiently, but quantum algorithms can, under widely believed complexity assumptions.

Some examples of learning separation based on cryptographic tasks were found. These famously include a learning problem based on the *discrete log* [20], which goes as follows. Given an n-bit prime number p and a generator a of the multiplicative group of integers modulo p, written as $\mathbb{Z}_p^*$, define for each fixed unknown integer $i \in [0, p-1]$ the function:

$$c_i : x \to (\log_a(x) \mod p) \in [i, i + \frac{p-3}{2}]\,. \tag{2.33}$$

The goal is to predict the labels of unseen outputs given access to a dataset of input-output pairs $(x_t, c_i(x_t))$ for T samples. Another notable learning exponential separation has been proven for Boolean functions [64], and a sub-exponential learning separation for a sequence modeling task was found in [65].

In [21], the authors propose that learning separations can be constructed based on BQP-complete problems. Consider a polynomially sized concept class (definition in Section 2.1) constituted of concepts that are all in BQP and at least one being BQP complete. Using a result from [66] that states that the learnability of a concept class implies the efficient evaluation of all the concepts, the considered concept class is hard to learn for a classical learner under widely believed complexity assumptions. It can also be learned efficiently with a quantum computer, by brute force, testing all possible concepts.

In [22], the authors propose a concept class that does not need to be polynomially sized to exhibit a quantum learning separation. We provide more details on how to define the size of a set of functions in Section 2.1.2. Consider a quantum state that depends on the input data $\rho(x)$, and an Observable O with an unknown sparse representation in the Pauli basis as follows

$$O(\alpha) = \sum_{i=1}^{m} \alpha_i P_i \,. \tag{2.34}$$

Define the functions $c_\alpha(x) = \mathrm{Tr}[\rho(x)O(\alpha)]$, and let them constitute the concept class $\mathcal{C} = \{c_\alpha\}_{\alpha\in[-1,1]^m}$. This concept class was proven to exhibit an exponential learning separation.

In Chapter 5, we will prove a quantum advantage for a concept class based on unknown Hamiltonian dynamics.

2.4. Bosonic Hamiltonians and Continuous Variables Quantum Computing

In Chapter 6 and Chapter 7 we will consider quantum circuits based on bosonic modes, with infinite-dimensional Hilbert spaces. In this section, we introduce the necessary background on bosonic Hamiltonians and continuous variable quantum information.

2.4.1. Continuous Variable Quantum Information

We start by introducing the central ideas of Continuous Variable Quantum Information [67]. In contrast to qubit-based computation, where observables can only take a finite discrete set of values upon measurement, Continuous Variable (CV) observables can take a value from an infinite number of values over a continuous interval. Quantum states can be expressed in terms of creation $\hat{a}^\dagger$ and annihilation $\hat{a}$ operators defined such that $\hat{a}\,|n\rangle = \sqrt{n}\,|n-1\rangle$, where $|n\rangle$ are the Fock states in a countably infinite-dimensional Hilbert space. The quadrature operators are the position operator $\hat{q}$ and its conjugate operator is the momentum operator $\hat{p}$, defined as follows,

$$\hat{q} = \frac{\hat{a} + \hat{a}^\dagger}{\sqrt{2}}\,, \tag{2.35}$$

$$\hat{p} = \mathbf{i}\frac{\hat{a} - \hat{a}^\dagger}{\sqrt{2}}\,. \tag{2.36}$$

Their commutation relation is given by $[\hat{q}, \hat{p}] = i$. The quadrature operators are self-adjoint and have a continuous spectrum with eigenvalues x spanning $\mathbb{R}$, and their corresponding eigenvectors $|x\rangle_{\hat{q}}$ for the position operator and $|x\rangle_{\hat{p}}$ for the momentum operator. These eigenvectors form a full basis of the Hilbert space:

$$I = \int_{-\infty}^{+\infty} |x\rangle_{\hat{q}} \langle x|_{\hat{q}}\, dx = \int_{-\infty}^{+\infty} |x\rangle_{\hat{p}} \langle x|_{\hat{p}}\ . \tag{2.37}$$

The conditions for a set of bosonic gates to be universal, that is, to be able to approximate any unitary transformation, were derived in [29]. The first set of gates of interest is the Gaussian gates, under which the set of Gaussian states is invariant. A Gaussian state is completely specified by its first and second moments of the quadrature operators. We provide more details on Gaussian states and Gaussian gates in Section 2.4.2. Gaussian gates applied to Gaussian states can be efficiently classically simulated [68] for a polynomial number of modes. This result is in some regard analogous to the Gottesman-Knill theorem on efficient simulation of Clifford circuits [12]. We present complexity results on the simulation of Gaussian circuits on exponentially many modes in Chapter 7. The generator of any Gaussian gate can be expressed as a quadratic polynomial in the quadrature operators (e.g. $\hat{p}^2 + \hat{q}$).

In order to reach universality, a non-Gaussian gate with a higher degree in quadrature operator has to be added to the model of computation [67], the most common non-Gaussian gates being either the cubic gate $e^{it\hat{q}^3}$ and the Kerr gate $e^{it\left(\hat{q}^2+\hat{p}^2\right)^2}$. These non-Gaussian gates are very difficult to implement in practice, and can therefore be considered “expensive”. For the purpose of Chapter 6, we choose our model of computation as being the following set of gates, defined by their corresponding generators:

- the Displacement Gate: $H = \hat{p}$
- the Squeezing Gate: $H = \hat{p}\hat{q} + \hat{q}\hat{p}$
- the Quadratic Gate: $H = \hat{p}^2$

- the Cubic Gate: $H = \hat{q}^3$
- the Beam-splitter: $H = \hat{q}_1\hat{p}_2 - \hat{q}_2\hat{p}_1$

2.4.2. Gaussian states

In what follows, we will consider systems composed of M bosonic modes (with $M = 2^n$ in Chapter 7). Let $\hat{a}_m^\dagger$ and $\hat{a}_m$, with $m = 1, \dots, M$, respectively denote the creation and annihilation operators for the m-th mode [67]. We consider the standard Hermitian *quadrature operators* as defined in Section 2.4.1, position $\hat{q}_m = \frac{1}{\sqrt{2}}(\hat{a}_m + \hat{a}_m^\dagger)$ and momentum $\hat{p}_m = \frac{i}{\sqrt{2}}(\hat{a}_m^\dagger - \hat{a}_m)$. They satisfy the canonical commutation relations $[\hat{q}_m, \hat{p}_{m'}] = i\delta_{mm'}$. Furthermore, in Chapter 7, we will focus on the case where an M-mode bosonic state ρ_0 evolves under the action of a Gaussian Bosonic (GB) circuit whose gates are generated by time-independent Hamiltonians that are quadratic in the position and momentum operators[2].

These GB generators, also known as free-bosonic generators, are arbitrary real-valued degree-two homogeneous polynomials on the quadrature operators

$$\hat{H} = \frac{1}{2}\hat{\vec{z}}^T K \hat{\vec{z}}, \text{ with } \hat{\vec{z}} = (\hat{q}_1, \dots, \hat{q}_M, \hat{p}_1, \dots, \hat{p}_M)^T , \tag{2.38}$$

where K is a real $2M \times 2M$ symmetric matrix. The vector $\hat{\vec{z}}$ allows us to express the commutation relations in the compact form $[\hat{\vec{z}}_\alpha, \hat{\vec{z}}_\beta] = i\Omega_{\alpha\beta}$, where $\Omega = iY \otimes I_M$. Here, Y is the usual 2×2 Pauli matrix and I_M the $M \times M$ identity matrix.

Next, we collect the expectation values of the position and momentum operators in a vector $\langle\hat{\vec{z}}\rangle = (\langle\hat{q}_1\rangle, \dots, \langle\hat{q}_M\rangle, \langle\hat{p}_1\rangle, \dots, \langle\hat{p}_M\rangle)^T \in \mathbb{R}^{2M}$, with $\langle\hat{x}\rangle$ the expectation value of $\hat{x}$ over ρ_0. As shown in the Supplemental Information (SI), evolving ρ_0 with a unitary generated by a GB generator $\hat{H}$ for a time t induces the evolution of $\langle\hat{\vec{z}}\rangle$ as

$$\frac{\partial\langle\hat{\vec{z}}\rangle}{\partial t} = \Omega K \langle\hat{\vec{z}}\rangle\,, \text{ so that } \langle\hat{\vec{z}}\rangle(t) = e^{t\Omega K}\langle\hat{\vec{z}}\rangle(0)\,. \tag{2.39}$$

Here, $\langle\hat{\vec{z}}\rangle(t)$ denotes the vector containing the expectation values of positions and momenta at time t, and thus $\langle\hat{\vec{z}}\rangle(0)$ represents the initial condition. Since the canonical commutation relations must be preserved, the propagator $e^{t\Omega K}$ is a $2M \times 2M$ symplectic matrix with real entries belonging to the Lie group $\mathbb{SP}(M, \mathbb{R})$ [67], which in turn implies that ΩK is an operator in the symplectic Lie algebra $\mathfrak{sp}(M, \mathbb{R})$.

---

[2] A generalization to time-dependent Hamiltonians is direct using standard Hamiltonian-simulation techniques [69].

Note that while, in general, $e^{t\Omega K}$ is not unitary, we characterize the quadratic Hamiltonians leading to unitary evolutions of the vector $\langle \hat{\vec{z}} \rangle$, when a gate generator is of the form $\hat{H} = \sum_{m,m'=1}^{M} h_{mm'} \, \hat{a}_m^\dagger \hat{a}_{m'} + \frac{\mathrm{Tr}[h]}{2} I_{2M}$, with h a Hermitian matrix. Then $[\Omega, K] = 0$, and the propagator $e^{t\Omega K}$ is the real-time evolution of the Hermitian $i\Omega K$. Hamiltonians of this form are known as *particle preserving* [70] since the hopping terms $\hat{a}_m^\dagger \hat{a}_{m'}$ move a boson from mode m' to mode m. We also characterize *non-particle-preserving Hamiltonians* of the form $\hat{H} = \sum_{m,m'=1}^{M} \Delta_{mm'}^\dagger \, \hat{a}_m \hat{a}_{m'} + \text{h.c.}$, whose propagator corresponds to the imaginary-time evolution of $\langle \hat{\vec{z}} \rangle$ under the effective Hamiltonian $-\Omega K$.

In addition to $\langle \hat{\vec{z}} \rangle$, we also collect the expectation value of products of quadrature operators over ρ_0 in the $2M \times 2M$ positive-definite covariance matrix $\vec{\sigma}$ whose entries are given by $\vec{\sigma}_{\alpha\beta} = \frac{1}{2} \langle \hat{\vec{z}}_\alpha \hat{\vec{z}}_\beta + \hat{\vec{z}}_\beta \hat{\vec{z}}_\alpha \rangle - \langle \hat{\vec{z}}_\alpha \rangle \langle \hat{\vec{z}}_\beta \rangle$ [71]. Analogously to Equation (2.39), we find (see the SI) that

$$\frac{\partial \vec{\sigma}}{\partial t} = \Omega K \vec{\sigma} - \vec{\sigma} K \Omega \,, \text{ so that } \vec{\sigma}(t) = e^{\Omega K t} \vec{\sigma}(0) e^{\mp \Omega K t} \,,$$

where the $-$ $(+)$ sign corresponds to particle (non-particle) preserving Hamiltonians, as defined above.

The previous insights pave the way to simulate the action of GB circuits on a gate-based quantum computer in Chapter 7.

2.5. Complexity Theory and Quantum Classes

Complexity theory provides a framework for classifying computational problems according to the resources required to solve them. It is fundamental for understanding the capabilities and limits of both classical and quantum computers. Two commonly studied classical complexity classes are:

- **P**: problems solvable by a deterministic Turing machine in polynomial time.
- **NP**: problems for which a candidate solution can be verified in polynomial time.

The question of whether P = NP is still open and lies at the heart of classical computational complexity.

Quantum computing introduced new models of computation, leading to a rich hierarchy of complexity classes capturing the power of quantum algorithms and quantum verification. This section defines the most relevant quantum classes, BQP, QMA, PostBQP, and DQC1 and explains their significance. But first we recall some basic definitions from complexity theory.

2.5.1. Inclusion, hardness, and completeness

A problem L is said to be *in* a complexity class $\mathcal{C}$ if there exists an algorithm in $\mathcal{C}$ that solves L correctly on all inputs.

A problem H is *$\mathcal{C}$-hard* if every problem $L \in \mathcal{C}$ reduces to H under polynomial-time reductions. In other words, an efficient solution for H would yield efficient solutions for all the problems in $\mathcal{C}$.

A problem is *$\mathcal{C}$-complete* if it is both in $\mathcal{C}$ and $\mathcal{C}$-hard. Such problems are the most representative of their class $\mathcal{C}$ and the hardest within it: solving any $\mathcal{C}$-complete problem efficiently implies solving all problems in $\mathcal{C}$ efficiently.

2.5.2. The class BQP

Definition 2.5 (BQP)
The class Bounded-error Quantum Polynomial time *(BQP) contains all decision problems solvable by a polynomial-size quantum circuit in polynomial time, with error probability at most* $1/3$ *for all instances. That is, for input x, the quantum algorithm outputs the correct answer with probability* $\geq 2/3$.

The choice of $1/3$ is arbitrary and could be any constant $< 1/2$: by repeating the algorithm and taking a majority vote, the probability of error can be made exponentially small. Examples of PromiseBQP-complete problems include:

- **Simulation of quantum circuits** Given a quantum circuit decide whether it outputs $|1\rangle$ with probability at least $2/3$ or at most $1/3$ when measuring the first qubit, promised that either is true.
- **Simulation of exponentially many coupled oscillators** as in [42].

2.5.3. The class QMA

The class *Quantum Merlin-Arthur* (QMA) generalizes NP to the quantum setting.

Definition 2.6 (QMA)
A decision problem L is in QMA if there exists a quantum verifier $|\psi\rangle$ running in polynomial-time such that:

- *For every $x \in L$, the quantum verifier accepts with probability $\geq 2/3$.*
- *For every $x \notin L$, all quantum proofs $|\psi\rangle$ are rejected with probability $\geq 2/3$.*

Here, Merlin provides the quantum proof, and Arthur verifies it with a quantum computer. Examples of QMA-complete problems include:

- **Local Hamiltonian problem**: Given a $k \geq 2$-local Hamiltonian $H = \sum_j H_j$, decide whether the ground state energy is below a or above b, promised that $b - a \geq 1/\text{poly}(n)$ [72].
- **Quantum k-SAT**: Decide whether there exists a quantum state satisfying all given quantum constraints [73].
- **Betti numbers**: Decide whether the k-th Betti number of a simplicial complex is zero or non-zero [74].

2.5.4. The class PostBQP

Definition 2.7 (PostBQP)
The class Post-selected BQP *(PostBQP) consists of problems solvable by a polynomial-time quantum algorithm that is allowed to condition on measurement outcomes (post-selection). Formally, the algorithm may discard runs where a certain measurement outcome does not occur, even if the probability of success is arbitrarily small.*

Remark: Aaronson showed that PostBQP coincides with the classical class PP (Probabilistic Polynomial time) [34], showing how much post-selection increases the computational power of quantum computers.

Examples of PostBQP-complete problems include:

- **Majority Boolean Formula Problem** Given a Boolean formula F on n variables, decide whether strictly more than half of all 2^n assignments satisfy F.

- **Simulation of post-selected quantum circuits** Given a quantum circuit post-selected on the first qubit measured to be $|1\rangle$ (promised that the probability is non-zero) decide whether it outputs $|1\rangle$ with probability at least $2/3$ or at most $1/3$ when measuring the second qubit, promised that either is true.

2.5.5. The class DQC1

Definition 2.8 (DQC1)
The class Deterministic Quantum Computation with One Clean Qubit *(DQC1) consists of decision problems solvable by a polynomial-time quantum algorithm that starts with only one qubit in a pure state (the "clean qubit") and all other qubits in the maximally mixed state. The algorithm then applies a polynomial-size quantum circuit and measures the clean qubit to determine the output.*

This model, introduced by Knill and Laflamme [75], was designed to probe the computational capabilities of highly noisy quantum systems. Despite its severe restrictions, essentially a quantum computer with only minimal quantum resources, it can efficiently solve certain problems that are believed to be classically intractable [76]. Positioned between P and BQP in computational power, DQC1 serves as a natural complexity class for describing near-term quantum devices operating with limited coherence and entanglement.

Examples of DQC1 problems include:

- **Normalized trace estimation**: Given a unitary matrix U (described as a quantum circuit), estimate $\mathrm{Tr}(U)/2^n$ to additive inverse polynomial accuracy.
- **Approximating the Jones polynomial at certain roots of unity**: restricted to links presented as the trace closure of braids [77].

Part I.

Designing PQC-based QML methods

CHAPTER 3

Gradients and frequency profiles of quantum re-uploading models

3.1. Introduction

Variational Quantum Algorithms (VQAs) have emerged as a prominent paradigm in the realm of quantum computing as a hybrid computational model suited for NISQ (Noisy Intermediate-Scale Quantum) [5] devices in conjunction with classical optimization techniques [4, 78]. These algorithms rely on the minimization of cost functions [79, 80], which encode specific computational problems. VQAs have been used to solve a variety of problems, including approximating ground states [6, 81, 82], combinatorial challenges [8], chemistry problems [83] and simulation of quantum systems [47, 84–87]. Furthermore, VQAs have served as quantum computing engines for tackling various machine learning (ML) tasks, such as function regression [88, 89], classification [90–92] or generative models [93–95]. We specifically make a distinction between linear models on the one side, introducing data either as input states or through encoding maps [96], and quantum re-uploading (QRU) schemes on the other side,

---

The contents of this chapter have been published in [35].

which introduce data iteratively throughout the execution of the quantum circuit [26, 49, 97, 98].

The performance of VQAs hinges on two critical properties: expressivity and trainability. Expressivity embodies the model's ability to represent precise solutions to the underlying problem, while trainability is a measure of the difficulty in finding the parameter set that yields the optimal attainable solution within the model. In the case of data-independent VQAs, expressivity can be intuitively understood as the proportion of attainable output states within the Hilbert space, quantified through closeness to t-designs [48]. In the context of data-dependent ML, expressivity pertains to the suitability of the output function in fitting the data [23, 24]. The universality of QRU models has been proven even with a single qubit [25, 26]. Trainability in VQAs, on the other hand, is closely linked to characteristics of the cost function, such as non-convexity [51] or vanishing gradients [11]. The relationship between the trainability of VQAs and QML schemes has been previously explored, in the absence of re-uploading [99]. Importantly, trainability and expressivity are usually mutually exclusive, and for VQAs in particular there exists a well-studied trade-off between these two properties [52, 100].

In this chapter, we specifically explore QRU models with a focus on exploring trainability and expressivity. Our investigation into trainability focuses on the on-average behavior of gradients, which can be related to the flatness of the cost function. We compare the cost functions of QRU models and base PQCs, which are circuits with the same architecture and observable as the QRU, but where the data gates are removed. The difference between the flatness of both cost functions is upper bounded by a quantity we refer to as *absorption witness*, which quantifies the influence of data gates on the quantum circuit when averaged over the parameter space. Such derivation opens a path to transfer existing knowledge about the flatness of PQCs [11, 101, 102] to guide the design of QRU models.

The second segment of our findings is related to the expressivity of data-dependent output functions generated by QRU models. It is known that any hypothesis function output by QRU models can be expressed as a generalized trigonometric polynomial, with the range of available frequencies contingent on the data encoding scheme [26, 103]. We show that, under reasonable assumptions, the average magnitude of individual frequency components in the hypothesis function rapidly tends to a Gaussian profile, with a variance scaling as $\sim \sqrt{L}$, with L being the number of re-uploading steps, while the support in frequencies scales as $\sim L$. This property inherently biases the attainable hypothesis functions as being heavily dominated by lower-frequency components. This has a direct consequence on the Lipschitz constants of these output functions.

This chapter is organized as follows. Section 3.2 delves into the expected norm of gradients in QRU models. Section 3.3 delves into the expressive capabilities of output functions within QRU models in terms of spectrum. Both sections are supported by numerical experiments showcasing agreement with our theoretical findings. Section 3.4 engages in a discussion of the implications and potential avenues opened up by our research. Conclusions are summarized in Section 3.5.

3.2. Gradients in QRU models

3.2.1. Losses for PQC and QRU

In this section, we focus on characterizing gradients of QRU models, as compared to those of PQCs. In the case of PQC, the gradients of interest are typically defined in relationship to their cost function $h_{\boldsymbol{\theta}}(0)$. However, the optimization of QRU models involves a cost function that depends on both the quantum circuit, expressed through $h_{\boldsymbol{\theta}}(x)$, and the available data, provided in pairs as $(x, y(x))$. Such cost function is usually given by averaging a distance between functions $\Delta(\cdot,\cdot)$ as

$$\mathcal{L}_X(\boldsymbol{\theta}) = \mathbb{E}_X\left(\Delta(h_{\boldsymbol{\theta}}(x), y(x))\right), \tag{3.1}$$

where $\mathbb{E}_X(\cdot)$ denotes expectation value over the training dataset $X = \{(x, y(x))\}$, usually composed by a discrete set of points. Notice that $\mathcal{L}_X$ is empirical as the training dataset is drawn from an unknown data distribution $X \sim \mathcal{D}$, and approximates the true unaccessible risk averaged over $\mathcal{D}$. In regression tasks, a common choice for the distance metric $\Delta(\cdot,\cdot)$ is the mean squared error.

Our interest lies in examining the gradients of the loss function of QRU models, expressed as

$$\partial_j \mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_X\left(\frac{\partial \Delta(h_{\boldsymbol{\theta}}(x), y(x))}{\partial h_{\boldsymbol{\theta}}(x)} \partial_j h_{\boldsymbol{\theta}}(x)\right). \tag{3.2}$$

The influence imposed by the choice of distance function $\Delta(\cdot,\cdot)$ can be readily bounded e. g. using its Lipschitz constant L_Δ. In particular,

$$\mathrm{Var}_{\boldsymbol{\Theta}}\left(\partial_j \mathcal{L}(\boldsymbol{\theta})\right) \leq L_\Delta^2 \, \mathrm{Var}_{\boldsymbol{\Theta}}\left(\mathbb{E}_X\left(\partial_j h_{\boldsymbol{\theta}}(x)\right)\right). \tag{3.3}$$

Therefore, we can bound vanishing gradients by studying only

$$\mathrm{Var}_{\boldsymbol{\Theta}}\left(\mathbb{E}_X\left(\partial_j h_{\boldsymbol{\theta}}(x)\right)\right) .$$

It is important to highlight that all results presented in this section are applicable for any distribution over parameters $\boldsymbol{\Theta}$.

3.2.2. Gradient of the loss function

We connect now the gradients of loss functions for QRU models and PQCs. First, the average of derivatives of hypothesis functions are zero, namely [52]

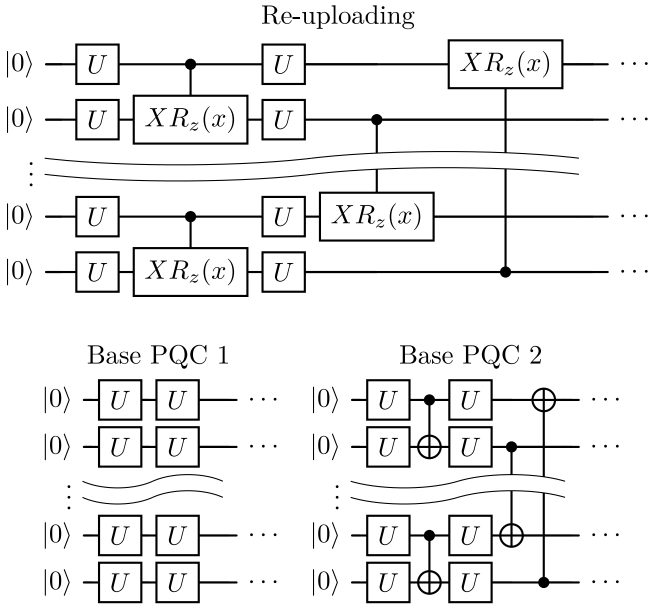
$$\mathbb{E}_{\boldsymbol{\Theta}}\left(\mathbb{E}_X\left(\partial_j h_{\boldsymbol{\theta}}(x)\right)\right) = \mathbb{E}_X\left(\mathbb{E}_{\boldsymbol{\Theta}}\left(\partial_j h_{\boldsymbol{\theta}}(x)\right)\right) = 0, \tag{3.4}$$

if the parameters $\boldsymbol{\theta}$ are sampled uniformly from $\boldsymbol{\Theta}$. As a consequence, due to the convexity of the square function, we have $\mathbb{E}\left(x\right)^2 \leq \mathbb{E}\left(x^2\right)$. By combining these two observations and the definition $\mathrm{Var}\left(x\right) = \mathbb{E}\left(x^2\right) - \mathbb{E}\left(x\right)^2$, we derive

$$\mathrm{Var}_{\boldsymbol{\Theta}}\left(\mathbb{E}_X\left(\partial_j h_{\boldsymbol{\theta}}(x)\right)\right) \leq \mathbb{E}_X\left(\mathrm{Var}_{\boldsymbol{\Theta}}\left(\partial_j h_{\boldsymbol{\theta}}(x)\right)\right), \tag{3.5}$$

which can be readily connected to $\mathrm{Var}_{\boldsymbol{\Theta}}\left(\partial_j \mathcal{L}(\boldsymbol{\theta})\right)$ via Equation (3.3). Therefore, we can bound the variance of cost functions in QRU by the average of variances cost functions of several PQC, defined by different fixed x_0. Bounds on $\mathrm{Var}_{\boldsymbol{\Theta}}\left(\partial_j h_{\boldsymbol{\theta}}(0)\right)$ have been studied in the context of PQC. In particular, BPs are defined for exponentially vanishing bounds to $\mathrm{Var}_{\boldsymbol{\Theta}}\left(\partial_j h_{\boldsymbol{\theta}}(0)\right)$ [11].

Equation (3.5) suggests that QRU models present vanishing gradients if the base PQC presents BPs, which means that it is recommendable to use architectures that avoid vanishing gradients such as in [57, 102] when designing QRUs. This statement is made from a purely trainability point of view, as it has been shown that such architectures are indeed classically simulable. We are only highlighting that given a QRU model, if removing the reuploading gates yields a PQC architecture known to suffer from vanishing gradients, then vanishing gradients will also affect the QRU architecture. In addition, Equation (3.5) does not guarantee non-vanishing gradients for QRU models derived from BP-free PQCs. As an example, consider a re-uploading model with parameterized single-qubit gates and data-encoding entangling gates arranged in an alternate-layered structure, measured by a sum of 1-local observables, as illustrated in Figure 3.1,



Re-uploading

Base PQC 1

Base PQC 2

Figure 3.1.: Quantum circuits for the first experiments. The re-uploading model is depicted on top, and compared to the PQCs described in the bottom line. The U gates here described correspond to arbitrary parameterized single-qubit operations. The circuit here described corresponds to one layer, and the depth is determined by the number of repetitions.

base PQC 1. A compatible base PQC is composed only of single-qubit parameterized gates. In this case, $h_{\boldsymbol{\theta}}(x_0)$ from the base PQC has large gradients [54]. The inclusion of data increases the accessible Hilbert space due to the presence of entangling operations, entanglement, which causes BPs [52].

Consider again the previous example, this time with a different base PQC which includes entangling gates, see Figure 3.1, base PQC 2. The gates U are considered distributed according to the Haar measure for single-qubit operations. In this new scenario, the PQC cost function $h_{\boldsymbol{\theta}}(0)$ suffers from BPs for sufficient depth [52, 54]. Notice that it is possible to decompose a data-dependent entangling gate as a fixed entangling gate and tunable single-qubit [104], allowing for introducing data through single-qubit operations. Intuitively, the data can be re-absorbed by the parameters to generate a new circuit with the same ansatz as the base

PQC. As a direct consequence, the gradients of the PQC and those of the QRU are of the same magnitude, $\mathbb{E}_X\left(\mathrm{Var}_{\boldsymbol{\Theta}}\left(\partial_j h_{\boldsymbol{\theta}}(x)\right)\right) \approx \mathrm{Var}_{\boldsymbol{\Theta}}\left(\partial_j h_{\boldsymbol{\theta}}(x_0)\right)$, for any x_0. This intuition motivates the newly coined concept of *absorption witnesses* in Definition 3.1 as the capability of the circuit to absorb the data into the parameters.

3.2.3. Absorption Witnesses

Before further expanding on absorption witnesses, it is convenient to introduce some auxiliary quantities in the context of QRU models. We take derivatives with respect to the j-th parameter. All operations preceding the j-th parameter (not included) are considered the right part of the circuit, operationally attached to the input state ρ_0. Operations that include and follow the j-th parameter are on the left side of the circuit, attached to the observable. This description is given by

$$\rho_j(\boldsymbol{\theta}_{R,j}, x) = U_{R,j}(\boldsymbol{\theta}_{R,j}, x)\ \rho_0\ U^\dagger_{R,j}(\boldsymbol{\theta}_{R,j}, x) \tag{3.6}$$

$$H_j(\boldsymbol{\theta}_{L,j}, x) = U^\dagger_{L,j}(\boldsymbol{\theta}_{L,j}, x)\ H\ U_{L,j}(\boldsymbol{\theta}_{L,j}, x). \tag{3.7}$$

The left/right parameters $\boldsymbol{\Theta}_{R/L,j}$ are assumed to be independent. For each of the right and left parts of the circuit, we can define the difference with respect to the reference data value $x = 0$ (corresponding to the PQC) as

$$B^{(t)}_{R,j}(\boldsymbol{\theta}_{R,j}, x; \rho_0) = \rho_j^{\otimes t}(\boldsymbol{\theta}_{R,j}, x) - \rho_j^{\otimes t}(\boldsymbol{\theta}_{R,j}, 0) \tag{3.8}$$

$$B^{(t)}_{L,j}(\boldsymbol{\theta}_{L,j}, x; H) = H_j^{\otimes t}(\boldsymbol{\theta}_{L,j}, x) - H_j^{\otimes t}(\boldsymbol{\theta}_{L,j}, 0) \tag{3.9}$$

We define the absorption witness as follows.

Definition 3.1 (Absorption witness)
Let $U(\boldsymbol{\theta}, x)$ be a re-uploading model as defined in Equation (2.21). *Let $U_{R/L,j}(\boldsymbol{\theta}, x)$ be the right and left parts of the circuit with respect to the j-th gate. The right/left absorption witnesses are*

$$\mathcal{B}^{(2)}_{R,j}(\rho_0) = \mathbb{E}_X\left(\left\|\mathbb{E}_{\boldsymbol{\Theta}_{R,j}}\left(B^{(2)}_{R,j}(\boldsymbol{\theta}_{R,j}, x; \rho_0)\right)\right\|_1\right), \tag{3.10}$$

$$\mathcal{B}^{(2)}_{L,j}(H) = \mathbb{E}_X\left(\left\|\mathbb{E}_{\boldsymbol{\Theta}_{L,j}}\left(B^{(2)}_{L,j}(\boldsymbol{\theta}_{L,j}, x; H)\right)\right\|_1\right). \tag{3.11}$$

The absorption witness defined above captures the effect of including data when averaging over the parameter space $\boldsymbol{\Theta}$. If $\mathcal{B}^{(2)}_{R,j}(\rho_0) = 0$, the input x yields an effect on $\rho_j(\boldsymbol{\theta}_{R,j}, x)$ equivalent to some change $\boldsymbol{\theta}_{R,j} \to \boldsymbol{\theta}^*_{R,j}$.

This effect is compensated when averaging over $\boldsymbol{\Theta}$. The logic is analogous to the left part of the circuit. As an illustrative example, assume a single-layer re-uploading model composed by applying any data-encoding layer after a PQC forming a t-design. By definition, t-designs approximate up to the t-th statistical moment of Haar measure and are thus insensitive (on average) to adding extra operations, in particular any operation given by data-encoding. However, this closeness to t-designs is no longer possible for ansatzes with (several) data-encoding gates interspersed between parameterized layers.

The absorption witnesses from Definition 3.1 bound the differences between variances for PQCs and QRU models as follows.

Theorem 3.1
Let $U(\boldsymbol{\theta}, x)$ be a re-uploading model as defined in Equation (2.21). *Then*

$$\begin{aligned} &\left|\mathbb{E}_X \left(\mathrm{Var}_{\boldsymbol{\Theta}} \left(\partial_j h_{\boldsymbol{\theta}}(x)\right)\right) - \mathrm{Var}_{\boldsymbol{\Theta}} \left(\partial_j h_{\boldsymbol{\theta}}(0)\right)\right| \\ &\qquad\qquad \leq 4\|V_j\|_\infty^2 \left(\|H\|_\infty^2 \, \mathcal{B}_{R,j}^{(2)}(\rho_0) + \, \|\rho_0\|_\infty^2 \, \mathcal{B}_{L,j}^{(2)}(H)\right) \end{aligned} \tag{3.12}$$

where ρ_0 is the initial state, H is the observable to measure, and $\mathcal{B}_{L,j}^{(2)}(H)$ and $\mathcal{B}_{R,j}^{(2)}(\rho_0)$ are the absorption witnesses from Definition 3.1.

The proof can be found in Section 3.6.1.

Computing the absorption witness is not easy, nor computationally efficient. Alternatively, it can be estimated by comparing variances of the magnitudes of gradients with and without data, as will be done in the numerical calculations of Section 3.2.5. Nevertheless, it provides a useful interpretation of the relationship between vanishing gradients for data-dependent QRU models, as compared to their base PQCs, where the BP phenomenon has been already studied [11, 101, 102]. The absorption witnesses quantify the expressivity difference between the PQC where the data uploading gates of the QRU are removed and that of the PQC where the data uploading gates are replaced by parameterized gates. As an illustrative example, consider an arbitrary Hamiltonian H, and a quantum re-uploading model composed of a gate $e^{iH\theta_0}$ immediately followed by a data uploading gate e^{iHx_0}. These gates can be combined as $e^{iH\theta_1}, \theta_1 = \theta_0 + x$. Since the relevant quantities are variances of, any shift of this kind does not affect the average behavior, yielding an absorption witness of exactly 0. On the other hand, a QRU model composed by two arbitrary Hamiltonians $e^{iH_1\theta}$, $e^{iH_2 x}$ does not admit a shift in θ to absorb x, yielding an absorption witnwess depending on $H_{1,2}$. This result is formulated in the same fashion as the ones in [52], extending their applicability to quantum machine learning.

Finally, note that it is in principle possible to construct pathological datasets such that the base PQC suffers from BP while the QRU model does not. In these cases, the data uploading gates need a careful design to cancel out the structures responsible for vanishing gradients. It is thus reasonable to assume that real-world datasets would not result in such behavior.

3.2.4. Gradients in layered QRU models

We consider in this section layered QRU models, in contrast to the results we presented earlier that apply to all QRU structures. In many practical scenarios there are several parameterized gates between each pair of encoding gates [101, 102]. An encoding gate and all preceding parameterized gates is referred to as a layer as

$$U(\boldsymbol{\theta}, x) = \prod_{l=1}^{L} V_l(x) u_l(\boldsymbol{\theta}_l). \tag{3.13}$$

In this representation, the parameterized gates $u(\boldsymbol{\theta}_l)$ are no longer defined by a single generator, and $\boldsymbol{\theta}_l$ is no longer one-dimensional. We can in this case study the absorption capability of each individual layer by defining the corresponding absorption witnesses as follows.

Definition 3.2 (Layerwise absorption witness)
Let $u(\boldsymbol{\theta}_l)$ be the l-th layer of a re-uploading model from Equation (2.21), and let $V(x)$ be the data-encoding operation applied immediately after $u(\boldsymbol{\theta}_l)$. The absorption witness for the l-th layer is

$$\mathcal{A}_l^{(t)} = \mathbb{E}_X \left(\left\| \mathbb{E}_{\boldsymbol{\Theta}_l} \left(V_l(x)^{\otimes 2} u(\boldsymbol{\theta}_l)^{\otimes 2} - u_l(\boldsymbol{\theta}_l)^{\otimes 2} \right) \right\|_1 \right). \tag{3.14}$$

We provide some examples where $\mathcal{A}_l^{(t)} = 0$. First, assume a data-encoding layer sharing the generator with the corresponding parameterized gates. In this case, we can read data-encoding as a simple shift of parameters $\boldsymbol{\theta}^* \to \boldsymbol{\theta} - x$ (recall that $\boldsymbol{\theta}$ is now multi-dimensional), and averages do not change as long as $\boldsymbol{\theta}$ is sampled uniformly. Another example is the case where the ansatz is composed by k-local 2-designs located in consecutively alternated qubits, as in [54], where any k-local data-encoding gates can be re-absorbed by definition.

The use of layered ansatzes and layerwise absorption witnesses allows for further simplifications of Theorem 3.1 by bounding the complete absorption witnesses.

Lemma 3.1
Consider a layered re-uploading model as in Equation (3.13)*. Then*

$$\mathcal{B}^{(2)}_{R,l+1}(\rho_0) \leq \mathcal{B}^{(2)}_{R,l}(\rho_0) + \|\rho_0\|^2_\infty \mathcal{A}^{(2)}_{l+1} \tag{3.15}$$

$$\mathcal{B}^{(2)}_{L,l}(H) \leq \mathcal{B}^{(2)}_{L,l+1}(\rho_0) + \|H\|^2_\infty \mathcal{A}^{(2)}_{l}. \tag{3.16}$$

The proof can be found in Section 3.6.2.

Consider now a layered circuit where $u_l(\cdot) = u_k(\cdot)$ for any pair (l, k). The previous result can be further simplified since $\mathcal{A}^{(2)}_l = \mathcal{A}^{(2)}$ for all values of l. Therefore

Corollary 3.1
For a layered re-uploading model, the absorption witnesses of large parts of the circuits can be bounded by absorption witnesses of small pieces, by

$$\mathcal{B}^{(2)}_{R,l}(\rho_0) \leq L\|\rho_0\|^2_\infty \mathcal{A}^{(2)}, \tag{3.17}$$

$$\mathcal{B}^{(2)}_{L,l}(\rho_0) \leq L\|H\|^2_\infty \mathcal{A}^{(2)}. \tag{3.18}$$

The proof follows by repeated application of Lemma 3.1, together with the observation $\mathcal{B}^{(2)}_{R,l}(\rho_0) = 0$ if no data-encoding layer is considered in the absorption witness. We can therefore give a simplified bound for the results from Equation (3.12) in the case of layered ansatz as

$$\begin{aligned} &|\mathbb{E}_X\left(\mathrm{Var}_{\boldsymbol{\Theta}}\left(\partial_j h_{\boldsymbol{\theta}}(x)\right)\right) - \mathrm{Var}_{\boldsymbol{\Theta}}\left(\partial_j h_{\boldsymbol{\theta}}(0)\right)| \\ &\qquad\qquad \leq 8L\|V_j\|^2_\infty \|H\|^2_\infty\ \|\rho_0\|^2_\infty \mathcal{A}^{(2)}. \end{aligned} \tag{3.19}$$

The result from Equation (3.19) is looser than Theorem 3.1, but easier to compute, since it depends only on the layerwise absorption witness $\mathcal{A}^{(2)}$ corresponding to shallow circuits.

3.2.5. Numerical results

In this section, we present our numerical results, focusing on the average gradient magnitudes of hypothesis functions generated by QRU models in comparison to base PQCs. This analysis serves to validate the findings presented in Theorem 3.1 regarding gradient variances and can be considered as a proxy for evaluating the absorption witnesses defined in Definition 3.1. We explore various ansatzes and use different data distributions for the experiments. Our code for these experiments is available in [105], and the data can be provided upon request.

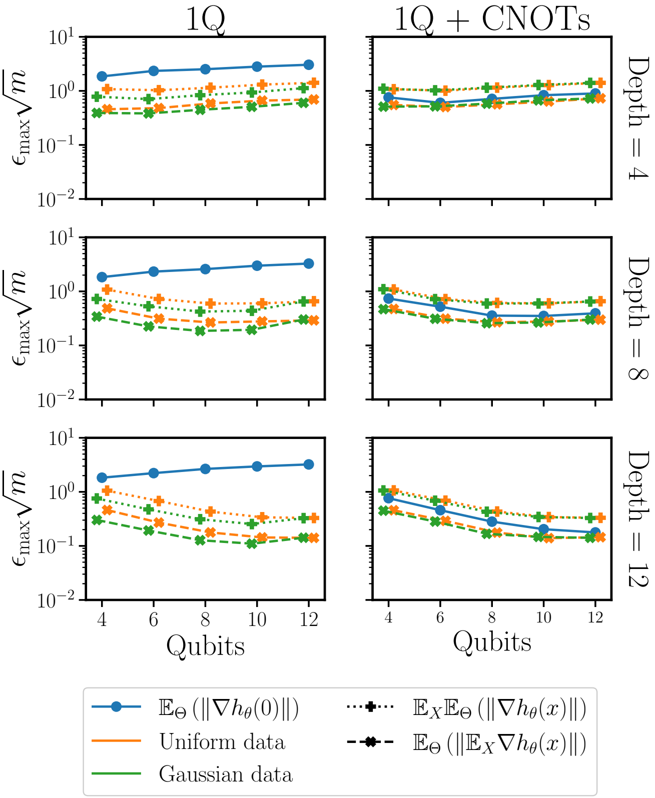


Figure 3.2.: Results for $\varepsilon_{\text{MAX}}\sqrt{m} \approx \mathbb{E}_X\left(\mathbb{E}_{\boldsymbol{\Theta}}\left(\|\nabla h_{\boldsymbol{\theta}}(x)\|\right)\right)$ (see Equation (3.20)) for QRU models with alternating layered ansatzes. Data is introduced through controlled-rotation gates. Parameterized gates are single-qubit arbitrary operations. Each row has an increasing depth in the circuit. The right column includes CNOT gates for the base PQC. In the left column, gradients follow different trends for the cases with and without data, implying data can not be re-absorbed into a reparametrization, in the sense of Theorem 3.1. In the right column, similar trends indicate large absorption capabilities.

For the numerical results we need to compute the magnitudes of the gradients on average. In order to reduce the computational complexity of this task, we will make use of the information content (IC) $I(\varepsilon)$ [106]. The IC is a statistical measure of the variability of the optimization landscape. In a nutshell, if $I(\varepsilon)$ is close to 1, then random displacements in $\boldsymbol{\theta}$ in the landscape change the value in $h_{\boldsymbol{\theta}}(x)$ in approximately ε, conveniently re-normalized by the norm of the displacement itself. The value $\varepsilon_{\mathrm{MAX}}$ at which $I(\varepsilon)$ is maximized serves as a numerical proxy for the average norm of the gradient, that is

$$\mathbb{E}_{\boldsymbol{\Theta}}\left(\|\nabla h_{\boldsymbol{\theta}}(x)\|\right) \sim \varepsilon_{\mathrm{MAX}}\sqrt{m}, \tag{3.20}$$

where $\sqrt{m}$ is the number of parameters. While this approximation is not capable of computing the exact value of $\mathbb{E}_{\boldsymbol{\Theta}}\left(\|\nabla h_{\boldsymbol{\theta}}(x)\|\right)$, it is robust against statistical fluctuations and provides reliable scalings. We refer the interested reader to Ref. [106] for an in-detail explaination of the validity and utility of IC to estimate gradients.

As a first example, we compare a re-uploading ansatz, consisting of single-qubit rotations and data-encoding entangling gates, with two different PQCs (see Figure 3.1). In both cases, we construct the hypothesis function measuring sums of single-qubit X Pauli measurements. The addition of entangling gates in PQC2 with respect to PQC1 is essential to exploring entangled states, and it plays a role in addressing the issue of vanishing gradients [54]. The data-encoding layer is a controlled operation $C-(XR_z(x))$, which can be absorbed into single-qubit and controlled rotations [104].

The results are shown in Figure 3.2. The columns correspond to the respective models (1-2), and the rows correspond to different depths of the ansatz. In the left column, corresponding to a PQC with only single-qubit gates, we observe a qualitative difference in the average gradients of the PQC and re-uploading circuits. The PQC is relatively insensitive to the number of qubits, as a consequence of the redundancy of multiple consecutive single-qubit gates. Adding data modifies this behavior. In the case of the entangling PQC (right column), the results without data align with previous works [54, 106]. The addition of data drawn from different distributions (Gaussian and uniform) introduces negligible differences in the results.

We turn our attention to translation-invariant ansatzes. These circuits are not capable of freely exploring the Hilbert space, but only its invariant subspace. This restriction reduces the freedom in these circuits, leading to an increase in the average gradients of the cost function for PQCs [102].

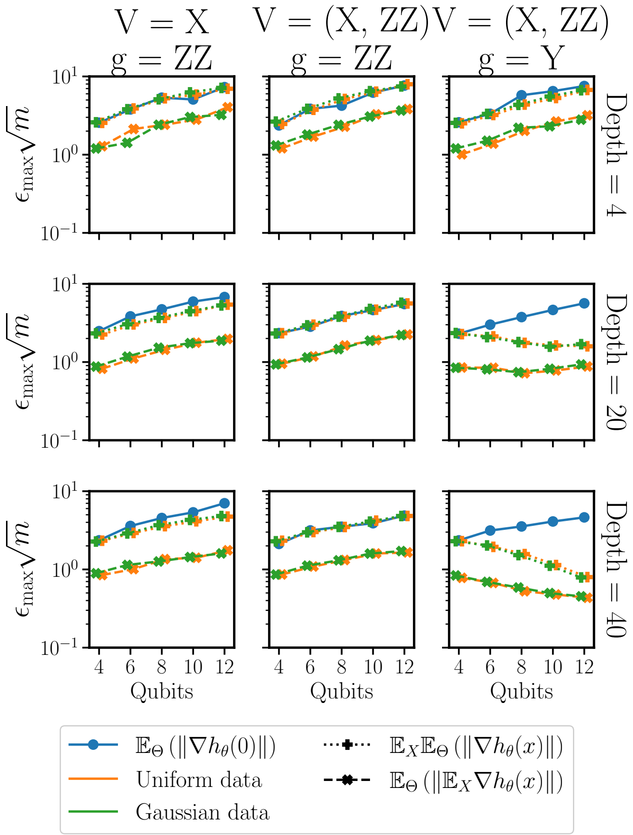


Figure 3.3.: Results for $\varepsilon_{\mathrm{MAX}}\sqrt{m} \approx \mathbb{E}_X\left(\mathbb{E}_{\boldsymbol{\Theta}}\left(\|\nabla h_{\boldsymbol{\theta}}(x)\|\right)\right)$ (see Equation (3.20)) for QRU models with translation-invariant layered ansatzes. Data is introduced through the generators g, and data through the generators V, indicated at the top for every column. Each row has an increasing depth in the circuit. In the left column, gradients follow approximately the same trends with and without data, implying high absorption capabilities in the sense of Theorem 3.1. For the middle column, the absorption is total since it can be done by a simple shift of parameters. The right column reveals different trends for the cases with and without data, implying low absorption.

We choose three layered models, based on the generators $X = \sum_q X_q$, $Y = \sum_q Y_q$, $ZZ = \sum_q Z_q Z_{q+1}$, where q cyclically iterates over all qubits. In the first model, the generator associated with parameters is $V_i = X$, and data-encoding is conducted through $g = ZZ$. The second model is given by $\{V\} = \{X, ZZ\}$, $g = ZZ$. The third model is defined by $\{V\} = \{X, ZZ\}$, $g = Y$. In all cases, the observable considered is X. Among these models, only the second one can automatically absorb data into parameters through shifts. Gaussian-distributed data is used in all cases.

Results are detailed in Figure 3.3. The columns correspond to the respective models, and the rows correspond to different circuit depths. For each model, the average norm of the gradient scales differently with the number of qubits, with and without data. Models 1 and 2 present similar behavior when including data. In particular, for model 2 results show no difference between the re-uploading model and PQC since the data can be perfectly re-absorbed through a simple shift. A significant difference is noticeable in the third model. In this case, the absence of BPs in all instances makes the QRU models trainable by construction.

3.3. Expressivity in QRU models

The hypothesis class of QRU can always be expressed as a generalized trigonometric polynomial [103], see Equation (2.28). In QRU models, the set of frequencies Ω is generated through the sequential Minkowski sum of the spectrum of the data encoding generators $\{\lambda_j\}_j$. In the general case, $\{\lambda_j\}_j$ consists of incommensurable real numbers, i. e. with non-rational ratios, and each new encoding step makes Ω combinatorially denser. In this section we first consider harmonic generators, i. e. with integer eigenvalues, and extend the results later to generic generators. As a main observation of this chapter, the behavior is similar in both cases.

3.3.1. Harmonic representation of quantum states

In this section, we introduce a representation of QRU models based on the Fourier decomposition of the hypothesis function. Such representation is useful for subsequent analytical results. Starting from Equation (2.21) and assuming the generators g_i possess an integer spectrum, we can express the state before measurement as

$$U(\boldsymbol{\theta}, x)\left|0\right\rangle = \sum_{k=-K}^{K} \sum_{j=1}^{2^n} c_{j,k}(\boldsymbol{\theta}) e^{i\mu k x} \left|j\right\rangle . \tag{3.21}$$

The coefficients $c_{j,k}$ form a matrix $\boldsymbol{C} \in \mathbb{C}^{2^n \times (2K+1)}$ that defines uniquely (up to a global phase) the output state of the re-uploading circuit before measurement. The matrix $\boldsymbol{C}$ depends only on the parameters $\boldsymbol{\theta}$ and the generators of the ansatz, but not on the data x. The value K corresponds to the largest attainable frequency, namely the sum of the largest eigenvalue for each generator used in the circuit. The recipe to construct $\boldsymbol{C}$ from the description of the circuit is detailed in Section 3.6.3. Notice this approach is equivalent to adding an extra dimension (frequency)to the standard brute-force state vector simulation, which is not efficient from a computational point of view. This harmonic representation of QRU models simulator is available on [105].

3.3.2. Vanishing high frequencies in QRU models

We use the above representations and the intuition that adding a data encoding layer corresponds to a convolution operation with the data encoding generator spectrum as defined below in Definition 3.3. For proof purposes, we assume that Haar random matrices are interleaved in between reuploading layers, as is common in most works exploring barren plateaus. We examine the statistical properties of the amplitude of the coefficients as a function of the frequency. We begin by defining the spectrum kernel of a harmonic Hermitian matrix.

Definition 3.3 (Harmonic spectrum kernel)
Let H be a $N \times N$ Hermitian matrix with integer eigenvalues $\{\lambda\}$ with multiplicities $m(\lambda)$. The spectrum kernel of H is the vector (indexed by k)

$$\mathcal{K}_H(k) = \begin{cases} m(k\mu)/N & \text{if } k\mu \in \{\lambda\} \\ 0 & \text{Otherwise} \end{cases}, \tag{3.22}$$

where μ is the largest value in $\mathbb{R}$ compatible with this description.

This function simply maps the eigenvalues of a Hermitian matrix into the normalized dimensionality of the corresponding eigenspace. For readability, we will refer to the *spectrum multiplicity function* simply as the *spectrum* for the remainder of the chapter.

In the case of layered QRU models, the spectra of their data-encoding generators and the number of layers L directly determine the set of attainable frequencies. The maximum attainable frequency is bounded by $L\|g\|_2$. We provide now some insight into how the coefficients are expected to behave.

Lemma 3.2 (Harmonic convolution)
Let $|\psi_{\boldsymbol{\theta}}(x)\rangle = U(\boldsymbol{\theta}, x)\,|\psi_0\rangle$ *be the output state of a re-uploading model, with data-encoded through the generator set* $\{g_j\}$*, each with spectrum* $\mathcal{K}_{g_j}$*, encoded as in Equation* (3.21)*. Assuming that each parameterized step is drawn from the Haar measure of unitaries, then* $\sum_j |c_{j,k}|^2$ *is a random variable satisfying*

$$\sum_j |c_{j,k}|^2 \sim \mathrm{Dir}\left(\left(\mathcal{K}_{g_1} * \ldots * \mathcal{K}_{g_j} * \ldots * \mathcal{K}_{g_L}\right)(k)\right), \tag{3.23}$$

where $\mathrm{Dir}(\alpha_1, \alpha_2, \ldots)$ *is the Dirichlet distribution [107] and* $*$ *denotes the convolution.*

The proof can be found in Section 3.6.4. The Dirichlet distribution is a family of probability distributions for multidimensional variables $\boldsymbol{x} \in [0,1]^N$, subject to $\|\boldsymbol{x}\|_1 = 1$. The Dirichlet distribution over N variables is fully described by N parameters $\alpha_i \in \mathbb{R}_{>0}$. Dirichlet is the multidimensional extension of the beta distribution. A detailed definition of the Dirichlet distribution and auxiliary results are given in Section 3.6.5. For completeness, we define convolution as

$$(f * g)(k) = \sum_{l=-\infty}^{\infty} f[k]g[l-k]. \tag{3.24}$$

In other words, this lemma gives statistical properties of the frequency content, expressed as the norm of the 2^n quantum vector corresponding to each frequency (seee Equation (3.21)) for QRU models composed of a sequence of data uploading gates interleaved with gates drawn from the Haar distribution. It states that the vector of frequency content follows a multidimensional distribution whose mean is the result of the successive convolution of the multiplicity kernels of the data encoding gates. It follows a Dirichlet distribution because all values are positive and sum up to one as per the normalization of a quantum state.

It is worth discussing the role of the Haar distribution in this result. First, choosing random unitaries allows us to scramble the inner quantum state in the QRU model at each step, thus transforming the QRU circuit into a random walk in the space of frequencies, where the parameters in Dirichlet only account for the number of paths leading to the same outputs. Second, random choices of unitaries is in alignment with other works exploring trainability and expressivity in VQAs [11, 52, 54], which rely on sampling unitaries from a t-design. The difference between the Haar distribution and a t-design is rather technical, since t-designs are sets of unitaries with the same statistical moments as the Haar measure, up to degree t [48]. With respect to Lemma 3.2, lowering the requirements in the parameterized steps from Haar distribution to t-design would imply to substitute the Dirichlet distribution with another probability distribution with the same t-statistical moments. Technical descriptions of this transformations are left as open questions for future research. Note that our results lose their validity if the parameterized steps are drawn with respect to other distributions of unitaries.

The previous result immediately implies the following.

Theorem 3.2 (Single-generator convolution)
Let $|\psi_{\boldsymbol{\theta}}(x)\rangle = U(\boldsymbol{\theta}, x)\,|\psi_0\rangle$ *be the output state of a re-uploading model, with data-encoded through the generator* g*, with spectrum* $\mathcal{K}_g$*, encoded as in Equation* (3.21)*. Assuming that each parameterized step is drawn from the Haar measure of unitaries, then*

$$\sum_j |c_{j,k}|^2 = \mathrm{Dir}\left(\left(\mathcal{K}_g^{*L}\right)(k)\right), \tag{3.25}$$

where $(\cdot)^{*L}$ *denotes the* L*-fold convolution.*

The proof is immediate from extending Lemma 3.2.

We provide two explicit examples to distinguish the cases captured by Lemma 3.2 and Theorem 3.2. Consider the single-qubit generator $g = (Z_0 + I)/2$, with spectrum $\mathcal{K}_g = (0, 1)$. To illustrate Lemma 3.2, we choose the list of generators as $\{2^l g\}_{l=0}^{L}$, yielding a convolution

$$\left(\mathcal{K}_{g_1} * \ldots * \mathcal{K}_{g_j} * \ldots * \mathcal{K}_{g_L}\right)(k) = 1, \forall k \in \{0, \ldots, 2^L - 1\}. \tag{3.26}$$

On the other hand, illustrating Theorem 3.2 we consider a repeated application of g, yielding

$$\left(\mathcal{K}_g^{*L}\right)(k) = \binom{k}{L}, \forall k \in \{0, \ldots, L\}. \tag{3.27}$$

The behavior in the two cases of the random variable $\sum_j |c_{j,k}|^2$ is significantly different. In the first case, the output is a flat distribution of exponential size. On the contrary, the second case is a distribution of linear size with high concentration in its mean values.

Notice that, provided that the data generator is known, it is possible to classically store $\mathcal{K}_g^{*L}$ within memory of size $\mathcal{O}\left(L\|g\|_2/\mu\right)$, with computational cost $\mathcal{O}\left((L\|g\|_2/\mu)^3\right)$. This allows us to classically characterize the frequency profile prior to executing the QRU model in quantum hardware for harmonic generators with only polynomially many eigenvalues.

The previous theorem can be readily interpreted in the limit of large L by virtue of the central limit theorem [108]. The repeated convolution of any random variable with a variance of σ and a probability distribution in the spaces L^1 and L^2, tends to a normal distribution in a weak sense. We can thus obtain the following result.

Corollary 3.2 (Vanishing high frequencies)
In the conditions of Theorem 3.2 and for large number of re-uploading L,

$$\lim_{L\to\infty} \sum_j |c_{j,k}|^2 \sim \mathrm{Dir}\left(\mathcal{N}\left(0, \sigma_g^2 L\right)(k)\right), \tag{3.28}$$

where σ_g is the standard deviation of the spectrum $\mathcal{K}_g$, and $\mathcal{N}(\mu, \sigma^2)$ is the normal distribution.

This observation implies that tails of the distribution vanish exponentially for large frequencies and there is a concentration in the low-frequency terms as the magnitudes of high-frequency terms vanish. In asymptotic scaling the available spectrum reduces from $\|g\|_\lambda^2 L$ to $\sigma_g\sqrt{L}$. For interpretability, recall the example $g = (Z_0 + I)/2$, yielding a binomial distribution in the convolution of the subsequent spectra. The binomial distribution rapidly tends to a gaussian distribution.

The results from Theorem 3.2 and Corollary 3.2 can be extended to non-harmonic generators. For readability, we postpone this result until Section 3.3.4.

The previous discussion considers the effects of the spectrum on the internal state of the re-uploading model in its harmonic representation. We are however not interested in the state itself, but rather in $h_{\boldsymbol{\theta}}(x)$ measured as an expectation value of this internal state. The Fourier components $h_{\boldsymbol{\theta}}(x)$ satisfy the following corollary.

Corollary 3.3
Let $|\psi_{\boldsymbol{\theta}}(x)\rangle$ be the output state of a re-uploading model, with a single data-encoding generator g with spectrum $\mathcal{K}_g$. Let $h_{\boldsymbol{\theta}}(x)$ be the hypothesis function induced by the observable H in the re-uploading model, as in Equation (2.27), and let $a_k(\boldsymbol{\theta})$ be their corresponding Fourier coefficients as

in Equation (2.28)*. In the conditions of Theorem 3.2, and for symmetric spectra* $\mathcal{K}_g(k) = \mathcal{K}_g(-k)$,

$$\|H\|_\lambda^{-2} |a_k(\boldsymbol{\theta})|^2 \leq p_k \tag{3.29}$$

$$p_k \sim \mathrm{Dir}(\mathcal{K}_g^{*2L}(k)), \tag{3.30}$$

where p_k *is a multidimensional probability distribution sampled from the Dirichlet distribution defined by the L-fold convoluted spectrum [107] as* $\mathrm{Dir}(\mathcal{K}_g^{*2L}(k))$.

Additionally, this result extends to the Gaussian distribution in the limit of large L as for Corollary 3.2.

The results from this section show that the frequency terms of $h_\theta(x)$ tend to follow a Gaussian profile of width $\sim L$, in the assumption that the generator of data-encoding gates is repeated in the QRU model. However, the frequency support of these functions scales linearly in L. As an immediate consequence, only frequencies $\omega \in \mathcal{O}(\sqrt{L})$ have practical support on average, while larger frequencies have exponentially vanishing weight in the hypothesis function. Note, that the Gaussian profile described by Corollary 3.2 does not imply a dense frequency space, which is still restricted to integer frequencies. This result holds even in the case where the generator provides exponentially-in-qubits many frequencies. It is then possible to have exponentially large frequency sets even with a small number of re-uploading steps, and the Gaussian approximation still holds with $\sigma \sim \sqrt{L}e^n$.

3.3.3. Lipschitz expressivity

In this section, we delve into a more practical understanding of the expressivity of hypothesis functions in terms of the magnitude of their derivatives. The ability to capture fine-grained data patterns depends on the function's ability to access high rates of change, i.e., the magnitude of its derivative. This concept can be quantified through the maximum value of the derivative, known as the optimal Lipschitz constant. For a function f, the optimal Lipschitz constant is defined as

$$\mathcal{L}(f) = \max_x |\partial_x f(x)| \,. \tag{3.31}$$

The Lipschitz constant is closely related to Fourier analysis, as high derivatives can only be achieved if the Fourier spectrum includes high frequencies with significant coefficients. Specifically,

$$\mathcal{L}(f) \leq \sum_{k=-K}^{K} |k|\mu|a_k e^{ik\mu x}|, \tag{3.32}$$

where a_k represents the Fourier coefficients of the hypothesis function.

We introduce an upper bound to the Lipschitz constant inspired by Equation (3.31), adapted for QRU models and properly normalized with respect to the measured observable as

$$\Lambda(h_{\boldsymbol{\theta}}) = \sum_{k=-K}^{K} \mu|k||a_k|. \tag{3.33}$$

It is straightforward to see that $\Lambda(h_{\boldsymbol{\theta}}) \geq \mathcal{L}(h_{\boldsymbol{\theta}})$, and therefore we are going to use this quantity as a proxy for it. For readability, this optimal Lipschitz constant upper bound will be referred to LB in the subsequent sections of this chapter and be noted $\Lambda(h_{\boldsymbol{\theta}})$ unless otherwise specified.

Using results from previous sections we study $\Lambda(h_{\boldsymbol{\theta}})$, starting with a result giving tight bounds on the asymptotic average of the LB over the parameters $\boldsymbol{\Theta}$. The results stated in the next and subsequent propositions stem from the conditions discussed in Section 3.3.2, namely tunable gates are drawn from the Haar distribution.

Theorem 3.3 (Average LB)
Let $h_{\boldsymbol{\theta}}(x)$ be the hypothesis function of a re-uploading model for which Theorem 3.2 applies. Let $\Lambda(h_{\boldsymbol{\theta}})$ be the LB as defined in Equation (3.33). Then,

$$\|H\|_{\lambda}\sqrt{2L}\mu\sigma_g \leq \lim_{L\to\infty} \mathbb{E}_{\boldsymbol{\Theta}}\left(\Lambda(h_{\boldsymbol{\theta}})\right) \leq \|H\|_{\lambda}\frac{4}{\sqrt{\pi}}\sqrt{L}\mu\sigma_g \tag{3.34}$$

The proof can be found in Section 3.6.7. Notice the tightness of the bounds above since $2\sqrt{2}/\sqrt{\pi} \approx 1.6$.

The following subsection quantifies the likelihood of the LB different from the average. Notice that values smaller than the average are not relevant due to the definition of $\Lambda(h_{\boldsymbol{\theta}})$. We can leverage the insights from previous results, particularly the role of Dirichlet distributions, to derive the following result:

Theorem 3.4 (Deviation of LB)
Let $h_{\boldsymbol{\theta}}(x)$ be the hypothesis function of a re-uploading model for which Theorem 3.2 applies, with data-encoding generator g. Let $\Lambda(h_{\boldsymbol{\theta}})$ be its LB as

defined in Equation (3.33)*. Then,*

$$\lim_{L\to\infty} \text{Prob}\left(\Lambda(h_{\boldsymbol{\theta}}) - \|H\|_{\lambda}\sqrt{2L}\sigma_g\mu \geq t\right) \in \mathcal{O}\left(\exp\left(-\frac{t^2}{\text{poly}(L\mu)}\right)\right). \quad (3.35)$$

The proof can be found in Section 3.6.8. Notice this result automatically bounds the probability of the optimal Lipschitz constant itself of being bigger than $\sqrt{2L\mu}\sigma_g$.

The previous theorem can be further refined to provide a tighter bound on the likelihood of large deviations from the LB. As mentioned in the detailed proof, when Theorem 3.2 holds the weight of each frequency and tends to follow a Gaussian-like profile, with central frequencies having exponentially larger probabilities than the extremal ones. It is expected that the primary contributions to $\Lambda(h_{\boldsymbol{\theta}})$ come from these central frequencies, which also have the smallest prefactors. Taking this into account, we can update the results from Theorem 3.4 to provide a more precise bound,

$$\lim_{L\to\infty} \text{Prob}\left(\Lambda(h_{\boldsymbol{\theta}}) - \|H\|_{\lambda}\sqrt{2L}\mu\sigma_g \geq t\right) \in \mathcal{O}\left(\exp\left(-\frac{t^2}{(\sigma_g\mu\sqrt{L})^3}\right)\right). \quad (3.36)$$

The vanishing high frequencies from Section 3.3.2 have consequences on the properties of the attainable hypothesis functions. In particular, its maximal derivative with respect x, given by the Lipschitz constant, scales in average with $\sqrt{L}$, and the probability of finding larger Lipschitz constants vanishes super-exponentially fast. This imposes in practice constraints on the capability of the hypothesis functions to capture fine details in the data, effectively restricting target functions that can be approximated by QRU models.

3.3.4. Extension to generic data generators

In previous subsections, we have proven the phenomenon of vanishing high frequencies and its consequences on the Lipshitz constant for harmonic data generators. In this subsection, we extend the results from Section 3.3.2 and Section 3.3.3 to any data generator. We start by defining the spectrum kernel for generic Hermitian matrices.

Definition 3.4 (Hermitian spectrum kernel)
Let H be a $N \times N$ Hermitian matrix with integer (positive or negative) eigenvalues $\{\lambda\}$ with multiplicities $m(\lambda)$. We define the vector $\vec{\mu} \in \mathbb{R}^D$, with $\mu_i/\mu_j \in \mathbb{R}\backslash\mathbb{Q}\ \forall (i,j)$, such that any eigenvalue can be written as $\lambda = \vec{\mu} \cdot \vec{k}$, with $\vec{k} \in \mathbb{Z}^D$. We refer to the number of anharmonic dimensions as $D \leq 2^n$, where n is the number of qubits. We define the spectrum kernel of H as $\mathcal{K}_H$ such that

$$\mathcal{K}_H(\vec{k}) = \begin{cases} m(\lambda)/N & \text{if } \vec{k} \cdot \vec{\mu} \in \{\lambda\} \\ 0 & \text{Otherwise} \end{cases} \tag{3.37}$$

Where each μ_j is the largest value in $\mathbb{R}$ compatible with this description.

We note the covariance of this spectrum as the $D \times D$ matrix Σ_g. The average of this spectrum is 0 since we consider traceless generators. The results from Lemma 3.2 and Theorem 3.2 hold in the non-harmonic case. In this scenario, the convolution must be done in a D-dimensional space, leading to D-dimensional frequency profiles. The convoluted spectrum, for the single-generator case, can be stored in a memory structure of size $\mathcal{O}\left((L\|g\|_2/\min_j \mu_j)^D\right)$. Notice that for each eigenvalue λ there exists only one compatible $\vec{k}$, due to the irrational ratios between elements in $\vec{\mu}$.

The central limit theorem still applies in the non-harmonic case as well, leading to the following result.

Corollary 3.4 (Vanishing high frequencies)
Given the conditions of Theorem 3.2 for non-harmonic generators and for large number of re-uploadings L,

$$\lim_{L\to\infty} \sum_j \left| c_{j,\vec{k}} \right|^2 \sim \mathrm{Dir}\left(\mathcal{N}\left(0, \Sigma_g L\right)\left(\vec{k}\right)\right). \tag{3.38}$$

Following the reasoning from the harmonic case, we focus now on the Lipschitz constant of the hypothesis functions. The definition from Equation (3.33) can be extended to

$$\Lambda(h_{\boldsymbol{\theta}}) = \sum_{\omega \in \Omega} |\omega||a_\omega|, \tag{3.39}$$

with $\omega = \vec{\mu} \cdot \vec{k}$. Since $\vec{k}$ has integer values and $\vec{\mu}$ has irrational ratios among its elements, there is at most one solution of $\vec{k}$ for each ω. With this definition, we can formulate results analogous to Theorem 3.3 and Theorem 3.4.

Corollary 3.5 (Lipschitz bounds for non-harmonic generators)
Let $h_{\boldsymbol{\theta}}(x)$ be the hypothesis function of a re-uploading model for which Corollary 3.4 applies. Let $\Lambda(h_{\boldsymbol{\theta}})$ be the LB as defined in Equation (3.39)*. Then,*

$$\lim_{L\to\infty} \mathbb{E}_{\boldsymbol{\Theta}}\left(\Lambda(h_{\boldsymbol{\theta}})\right) \leq \frac{4}{\sqrt{\pi}} \|H\|_\lambda \sqrt{\mathrm{Tr}(\Sigma)}\ \|\vec{\mu}\|_2 \sqrt{L} \tag{3.40}$$

$$\lim_{L\to\infty} \mathbb{E}_{\boldsymbol{\Theta}}\left(\Lambda(h_{\boldsymbol{\theta}})\right) \geq \sqrt{2}\ \|H\|_\lambda \sqrt{\min_\lambda(\Sigma)}\, \|\vec{\mu}\|_2 \sqrt{L}. \tag{3.41}$$

The proof of Corollary 3.5 can be found in Section 3.6.9. In addition, following the same reasoning leading to Theorem 3.4, we can infer exponential concentrations of $\Lambda(h_{\boldsymbol{\theta}})$ around its average values, by

$$\lim_{L\to\infty} \mathrm{Prob}\left(\Lambda(h_{\boldsymbol{\theta}}) - \|H\|_\lambda \sqrt{2L} \|\vec{\mu}\|_2 \sqrt{\min_\lambda(\Sigma)} \geq t\right) \\ \in \mathcal{O}\left(\exp\left(-\frac{t^2}{\left(\sqrt{\max_\lambda(\Sigma_g)}\max_j(\mu_j)\sqrt{L}\right)^3}\right)\right). \tag{3.42}$$

In light of the previous theorem, we can observe that the vanishing high frequencies phenomenon extends to non-harmonic generators, with minor changes with respect to the harmonic case, rooting from norm bounds in the multi-dimensional space. The tightness of these bounds depends on the regularity of the anharmonic spaces, which is reflected into the values of $\vec{\mu}$ and the eigenvalues of Σ.

An immediate consequence of this section is that QRU models can have a dense frequency spectrum without significantly modifying the envelope of the frequency profile. The only elements of QRU models allowing to increase the set of available frequencies in practice are L and the spectrum profile Σ, while $\vec{\mu}$, which can be related to $\|g\|_\lambda$ have a more modest effect. It is possible to reach exponentially many different frequencies by using generators with exponentially large $\mathcal{K}_g$.

The number of different frequencies directly affects the surrogability of the studied QML models In the case the frequency space is polynomial in the number of qubits, it is possible to construct a classical model fitting the corresponding generalized trigonometric polynomial [109]. On the other hand, exponentially large frequency spaces do not admit arbitrary efficient classical representations. The findings detailed in this chapter provide methods to circumvent surrogability. This can come from generators with exponentially large spectra, or designed in such a way that the frequency space scales exponentially with L, for instance with convolutions of highly non-harmonic spectra.

3.3.5. Numerical results

In this section, we show the results of a series of numerical experiments in which such conditions are relaxed and show that the theoretical results still apply. We use three models to test different situations. The first two models are constructed with permutation-invariant generators, which correspond to PQC that have been proven to be trainable [57]. Those models express only the symmetric subspace in the available Hilbert space. In the first model (A), $g = X, V = ZZ$, and the second model (B) $g = X, V = \{Y, X, ZZ\}$. For the third model, $g = X$, and the parameterized pieces are sampled from the Haar measure, that is the set V is free. We choose these models to have full control of the spectrum of the generator $\mathcal{K}_{g=X}$, which allows us to informatively compare to the theorems. All experiments were conducted with systems of 4 qubits unless explicitly stated, without affecting the scaling of the obtained results.

The first experiment tests Theorem 3.2 and Corollary 3.2, and the results can be seen in Figure 3.4. In model (A), the spectrum spreads towards large frequencies with the number of data re-uploadings. The parameterized gates are not general enough to support the theoretical results. For models (B) and (C), the Gaussian limit is matched even for a moderately small number of layers. This means that even though the theorem is proven for Haar random unitaries, the vanishing high-frequencies behavior still holds for model (B), even though the Haar condition is not guaranteed. Notice the difference in spreads for models (B) and (C). This is a consequence of the space explored by the ansatz. Model (B) is composed of a permutation invariant ansatz, and it is as general as possible only in the symmetric subspace, of dimension $n+1$. In this scenario, the spectrum of the corresponding g is flat (see the results for 1 layer in Figure 3.4), and the spread depends on the number of qubits n as $\sigma_g = \mathcal{O}(n)$. For the model (C), the spectrum of g with no restriction follows a binomial distribution, centered in $k = 0$, with $\sigma_g \in \mathcal{O}(\sqrt{n})$. A comparison between the theoretical and observed variances can be found in Figure 3.5, showing high agreement with the theoretical results.

We numerically check Theorem 3.4 in Figure 3.6. We depict in this figure the observed cumulative distribution functions (CDF) of both the numerically found LB, and $\Lambda(h_{\boldsymbol{\theta}})$ defined in Equation (3.31). These CDFs are compared to the upper bound from Theorem 3.4. Results show agreement with Theorem 3.4, and even indicate the possibility of finding tighter bounds, at least in terms of prefactors.

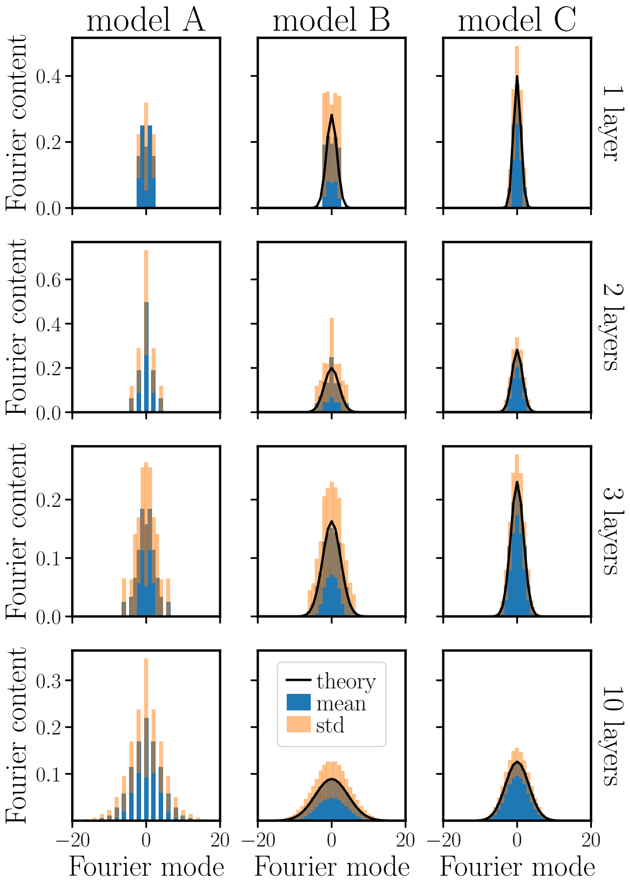


Figure 3.4.: Evolution of frequency spectrum with the number of layers, for models (A, B, C) detailed in the first paragraph of the section. The Fourier content refers to (average and standard deviation of) $|c_k|^2$. Model (A) is not general enough to follow Theorem 3.2, but still, a spread in frequencies is observed. Model (B) is permutation-invariant and almost fully general in the symmetric space, and model (C) is general with no restrictions in the Hilbert space. As L increases, the Fourier spectrum approximates a Gaussian profile with increasing variance according to Equation (3.28). The values σ_g for models (B) and (C) change due to the constraint in the available space.

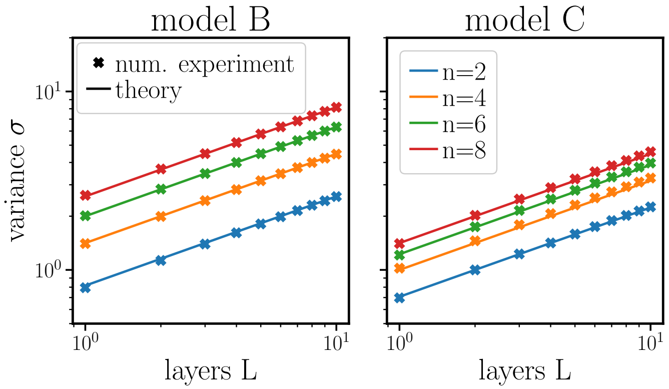


Figure 3.5.: Variances for the tending-to-Gaussian profiles from Figure 3.4, for different numbers of qubits, in the y-axis, while the x-axis corresponds to the number of layers. The left figure corresponds to model (B), with $\sigma_g^2 = n(n+2)/12$. The right figure corresponds to model (C), with $\sigma_g^2 = n/4$. The scaling in $\sqrt{L}$ is in agreement with Equation (3.28).

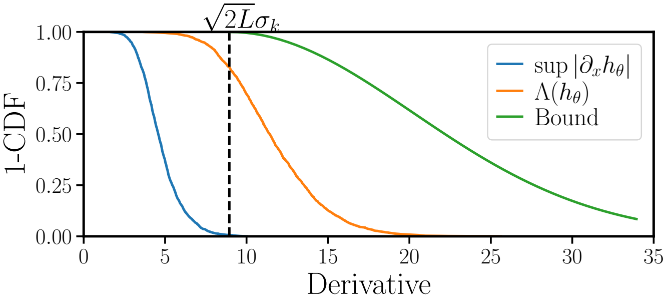


Figure 3.6.: Inverse CDFs for the optimal Lipschitz constants and $\Lambda(h_{\boldsymbol{\theta}})$, as compared to the bounds from Theorem 3.4. The x-axis indicate the value for each CDF, respectively $\sup_x |\partial_x h_{\boldsymbol{\theta}}(x)|$, $\Lambda(h_{\boldsymbol{\theta}})$ and t for each line.

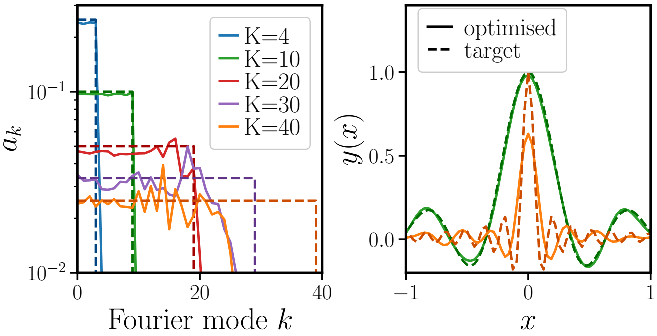


Figure 3.7.: Time and frequency domains representations of the functions of circuits (B) trained to match increasingly sharp cardinal sinus that yields increasing high-frequency content. From Fourier mode $k = 20$, the model is not able to match the amplitude, exhibiting a consequence of vanishing high frequencies.

Training

All the LB results describe an average behavior for $\boldsymbol{\Theta}$. In this subsection, we briefly explore the effect of previous results in the training. We task model (B) to learn functions whose Fourier coefficients follow a step function of increasing width. This approximately corresponds to a cardinal sinus of decreasing width.

We display results of trained QRU models in Figure 3.7. In the top figure, we show the Fourier components of different functions to be fitted (in dashed lines), and the hypothesis functions after training (solid lines). The target function is learned by the model for $K \leq 20$ but the hypothesis function fails to capture high frequencies from $k > 25$. Notice that the obtained hypothesis functions for $K = \{30, 40\}$ seem to saturate the expressivity capabilities of the model. The bottom figure represents the functions in the data domain for $K = \{4, 40\}$.

3.4. Discussion

We turn our attention first to gradients. From our results, we can infer that vanishing gradients are avoided in QRU models if the base PQC is BP-free, and data can be absorbed into the parameterized gates. We can refer to existing literature on avoiding BPs for PQCs by restricting the dimensionality of the search space, by means of the dynamical Lie algebra [57, 102]. In a nutshell, the Lie algebra depends on the generators of the quantum model. Absorption witnesses can only be maintained close to 0 if the base PQC and the derived QRU model share a common Lie algebra. This observation allows one to choose data-encoding generators avoiding the emergence of BPs.

The average of $\Lambda(h_{\boldsymbol{\theta}})$ is a consequence of the vanishing high frequencies behavior that grows as $\sim \sigma_g \sqrt{L}$, as imposed by the central limit theorem. Deviating from this average is exponentially unlikely, as proven in Theorem 3.4. As discussed later, it is in principle possible to amplify high-frequency components, at the expense of losing all degrees of freedom in the process. Therefore, for practical purposes, we need to adjust the number of re-uploading layers according to the scaling $\sim \sqrt{L}$, and not $\sim L$, as suggested by other theoretical works on expressivity via generalization bounds [103].

In this chapter we derived the scaling of the Fourier spectrum of hypothesis functions with the number of layers, but not with the number of qubits n. Our numerical simulations focus on frequency spaces increasing polynomially with n. However, it is possible to construct data-generators with exponentially many equally probable different accessible frequencies [110]. In this scenario, Theorem 3.2 still holds, leading to a Gaussian profile of frequencies with variance $\sigma \in \Theta(2^n \sqrt{L})$. Note that exponentially many frequencies require exponentially many tunable parameters to match the number of degrees of freedom. Therefore, data-encoding generators with $\sim e^n$ different frequencies can only aim to efficiently learn functions with sparse Fourier representations, i.e. with only $\mathcal{O}(\text{poly}(n))$ non-zero Fourier coefficients in a $\mathcal{O}(e^n)$ frequency space.

The expressivity results from Section 3.3 imply direct limitations in the attainable hypothesis functions, but also give an intuition on how to amplify high-frequency Fourier components, or in other words how to maximize the Lipschitz constant. The only recipe to obtain a high-frequency Fourier profile is by repeatedly amplifying the eigenvectors corresponding to extreme eigenvalues of the data-encoding generator. This yields an extremal case as far as possible from the average case where unitaries are sampled from the Haar distribution. Without loss of generality, we may choose the ground state. The first step of the circuit would have to transform the initial state into the ground state of the data-encoding generator. For k-local Hamiltonians with $k \geq 2$, this problem is QMA-complete [72], and finding the hypothesis function with maximum high-frequency content implies repeatedly solving a QMA-complete problem. An example is choosing the data generator to be the

Hamiltonian of a transverse-field Ising model, constructed on an arbitrary graph. Such PQC does not suffer from BPs if the parameters respect the permutation invariance of the graph [102]. Therefore, reaching the hypothesis function with maximized high-frequency content is in general hard. A notable exception appears in layered circuits with one g, and $W_i = I$, where setting all parameters $\theta = 0$ suffices to maintain the quantum state aligned with the ground state of g. Notably, maximizing the Lipschitz constant in the experiments from Section 3.3.5 is feasible.

We have seen that hypothesis functions produced by layered QRU models have naturally vanishing high-frequency components, therefore limiting their Lipschitz constant. Regularization of the Lipschitz constant yields increased generalization and robustness of classical Neural Networks [111, 112]. As a consequence, this could hint toward a better generalization capacity of QRU models, in agreement with existing literature [113].

The scope of this chapter is the average behavior of QRU models. It shows concentration properties, similar to other existing results [11, 14, 114], and provides useful insights on the internal working principles of the model. This interpretability will be useful to develop QML models with specific properties. It will be possible to investigate protection against dequantization through peaked generalized trigonometric polynomials, in alignment with peaked circuits [115]. Restrictions of generators and parameterized steps in the circuits can be applied to constraint the behavior of output models, allowing for systematic exploration of ingredients in QRU.

3.5. Conclusions

We have explored the features of QRU models to understand the implications of injecting data into the better-studied PQCs models. Two main features are studied, first the magnitude of the gradients of the loss function, and second the frequency profile of the hypothesis function output by QRU models. Results were proven analytically and extended to more practical scenarios numerically.

We give analytical bounds for the connection between the variance of gradients in the hypothesis functions of QRU models and the cost functions on corresponding PQCs. Vanishing gradients of hypothesis functions imply vanishing gradients for any cost function to train ML models with, thus preventing trainability. The difference between QRU models and PQCs can be quantified by measuring the effect of adding data to the circuit, averaging over the parameter space. This is quantified by the coined concept of *absorption witness*. If data can be re-absorbed in a shift of parameters, then the gradients for PQCs and QRU models take similar value ranges. Results can be further simplified for the case of layered ansatzes. These results provide insights into the construction of QRU models protected against the phenomenon of BP, by using existing knowledge on PQC that do not exhibit BPs.

We prove also that QRU models suffer from vanishing high frequencies. Each additional data encoding operation corresponds to an additional convolution of the current spectrum with the data generator spectrum. As the number of layers, denoted as L, becomes large the span of attainable frequencies grows as $\mathcal{O}(L)$. However, the central limit theorem dictates that the frequency profile follows a Gaussian distribution, spreading out proportionally to $\sqrt{L}$. Therefore, in practice, only frequencies (approximately) bounded by $\sqrt{L}$ are available, with the contribution of higher frequencies exponentially vanishing. The vanishing high frequencies have direct consequences on the class of functions attainable by the QRU models. The average of the optimal Lipschitz constant scales with $\sqrt{L}$, exhibiting an exponentially decaying probability of exceeding this value. These findings offer insights into the inherent limitations of expressivity in QRU models and provide tools for estimating the computational resources required to represent specific datasets effectively.

The results derived in this chapter broaden our understanding of the properties of QRU models and provide guidelines for the design of re-uploading schemes. As an example, the concept of absorption witness can be employed to select generators ensuring an ansatz with the necessary characteristics to be trainable. For expressivity, adjusting the depth of the model can strike a balance between capturing intricate details in the data and avoiding overfitting. Consequently, we anticipate that these tools and insights will contribute to enhancing the applicability and performance of QRU models.

3.6. Proofs

3.6.1. Proof of Theorem 3.1

We begin by explicitly writing the derivatives of the hypothesis function

$$\partial_j h_{\boldsymbol{\theta}}(x) = \mathrm{Tr}\{U_{R,j}(\boldsymbol{\theta}_{R,j}, x)\ \rho_0 U^\dagger_{R,j}(\boldsymbol{\theta}_{R,j}, x) \left[V_j, U^\dagger_{L,j}(\boldsymbol{\theta}_{L,j}, x) H U_{L,j}(\boldsymbol{\theta}_{L,j}, x)\right]\}. \tag{3.43}$$

In this equation, the indices R, L indicate all operations before and after the $j-$th operation. We redefine the quantities for readability

$$\rho_j(\boldsymbol{\theta}_{R,j}, x) = U_{R,j}(\boldsymbol{\theta}_{R,j}, x)\ \rho_0\ U^\dagger_{R,j}(\boldsymbol{\theta}_{R,j}, x) \tag{3.44}$$

$$H_j(\boldsymbol{\theta}_{L,j}, x) = U^\dagger_{L,j}(\boldsymbol{\theta}_{L,j}, x)\ H\ U_{L,j}(\boldsymbol{\theta}_{L,j}, x) \tag{3.45}$$

The variance of these derivatives over $\boldsymbol{\Theta}$ is given by

$$\mathbb{E}_X \left(\mathrm{Var}_{\boldsymbol{\Theta}} \left(\partial_j h_{\boldsymbol{\theta}}(x)\right)\right) = \mathbb{E}_{\boldsymbol{\Theta}} \left(\mathrm{Var}_X \left(\partial_j h_{\boldsymbol{\theta}}(x)\right)\right) = \mathbb{E}_{\boldsymbol{\Theta}_{R,j}} \left(\mathbb{E}_{\boldsymbol{\Theta}_{L,j}} \left(\mathbb{E}_X \left(\left(\partial_j h_{\boldsymbol{\theta}}(x)\right)^2\right)\right)\right), \tag{3.46}$$

where we assume no correlation between the parameters in the left and right parts of the circuit. By calling the property $\mathrm{Tr}\{A \otimes B\} = \mathrm{Tr}\, A\, \mathrm{Tr}\, B$, we can plug Equation (3.43) into Equation (3.46) to obtain

$$\mathrm{Var}_{\boldsymbol{\Theta}, X} \left(\partial_j h_{\boldsymbol{\theta}}(x)\right) = \mathbb{E}_{\boldsymbol{\Theta}_{R,j}} \left(\mathbb{E}_{\boldsymbol{\Theta}_{L,j}} \left(\mathbb{E}_X \left(\mathrm{Tr}\left\{\rho_j(\boldsymbol{\theta}_{R,j}, x)^{\otimes 2} \left[V_j, H_j(\boldsymbol{\theta}_{L,j}, x)\right]^{\otimes 2}\right\}\right)\right)\right) \tag{3.47}$$

We aim to describe this quantity in terms of the difference between the QML models, partially described by data x, and their corresponding PQC models, where $x = 0$. We define the corresponding difference operators

$$B^{(t)}_{R,j}(\boldsymbol{\theta}_{R,j}, x; \rho_0) = \rho_j^{\otimes t}(\boldsymbol{\theta}_{R,j}, x) - \rho_j^{\otimes t}(\boldsymbol{\theta}_{R,j}, 0) \tag{3.48}$$

$$B^{(t)}_{L,j}(\boldsymbol{\theta}_{L,j}, x; H) = H_j^{\otimes t}(\boldsymbol{\theta}_{L,j}, x) - H_j^{\otimes t}(\boldsymbol{\theta}_{L,j}, 0). \tag{3.49}$$

We rearrange the terms in the integrand of Equation (3.47) as

$$\mathrm{Tr}\Big\{\rho_j(\boldsymbol{\theta}_{R,j},x)^{\otimes 2}\,[V_j,H_j(\boldsymbol{\theta}_{L,j},x)]^{\otimes 2}\Big\} = \tag{3.50}$$

$$\mathrm{Tr}\{\rho_j(\boldsymbol{\theta}_{R,j},0)^{\otimes 2}[V_j,H_j(\boldsymbol{\theta}_{R,j},x)]^{\otimes 2}\}+ \tag{3.51}$$

$$\mathrm{Tr}\Big\{B^{(2)}_{R,j}(\boldsymbol{\theta}_{R,j},x;\rho_0)[V_j,H_j(\boldsymbol{\theta}_{L,j},x)]^{\otimes 2}\Big\} = \tag{3.52}$$

$$\mathrm{Tr}\{H_j(\boldsymbol{\theta}_{R,j},x)^{\otimes 2}[\rho_j(\boldsymbol{\theta}_{R,j},0),V_j]^{\otimes 2}\}+ \tag{3.53}$$

$$\mathrm{Tr}\Big\{B^{(2)}_{R,j}(\boldsymbol{\theta}_{R,j},x;\rho_0)[V_j,H_j(\boldsymbol{\theta}_{L,j},x)]^{\otimes 2}\Big\} = \tag{3.54}$$

$$\mathrm{Tr}\{\rho_j(\boldsymbol{\theta}_{R,j},0)^{\otimes 2}[V_j,H_j(\boldsymbol{\theta}_{R,j},0)]^{\otimes 2}\}+ \tag{3.55}$$

$$\mathrm{Tr}\Big\{B^{(2)}_{R,j}(\boldsymbol{\theta}_{R,j},x;\rho_0)[V_j,H_j(\boldsymbol{\theta}_{L,j},x)]^{\otimes 2}\Big\}+ \tag{3.56}$$

$$\mathrm{Tr}\Big\{B^{(2)}_{L,j}(\boldsymbol{\theta}_{L,j},x;H)[\rho_j(\boldsymbol{\theta}_{R,j},0),V_j]^{\otimes 2}\Big\} \tag{3.57}$$

by recalling the identities $\mathrm{Tr}\{A^{\otimes 2}\} = \mathrm{Tr}\{A\}^2$ and

$$\mathrm{Tr}\{A[B,C]\} = \mathrm{Tr}\{B[C,A]\} = \mathrm{Tr}\{C[A,B]\}\,,$$

The term in Equation (3.55) corresponds to the standard variance in PQC. We denote it simply as $\mathrm{Var}_{\boldsymbol{\Theta}}\left(\partial_j h_{\boldsymbol{\theta}}(0)\right)$.

We move our attention now to Equation (3.56). This term measures the difference between QRU models and PQC in the right part of the quantum circuit. Assuming that the right and left parameters are uncorrelated, we can rewrite

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{\Theta}_{R,j}}\left(\mathbb{E}_{\boldsymbol{\Theta}_{L,j}}\left(\mathbb{E}_X\left(\mathrm{Tr}\Big\{B^{(2)}_{R,j}(\boldsymbol{\theta}_{R,j},x;\rho_0)\,[V_j,H(\boldsymbol{\theta}_{L,j},x)]^{\otimes 2}\Big\}\right)\right)\right) = \\ &\quad \mathrm{Tr}\Big\{\left(\mathbb{E}_X\left(\mathbb{E}_{\boldsymbol{\Theta}_{R,j}}\left(B^{(2)}_{R,j}(\boldsymbol{\theta}_{R,j},x;\rho_0)\right)\right)\right)\left[V_j,\mathbb{E}_{\boldsymbol{\Theta}_{L,j}}\left(H(\boldsymbol{\theta}_{L,j},x)\right)\right]^{\otimes 2}\Big\}\end{aligned} \tag{3.58}$$

Using von Neumann's trace and Hölder inequalities, with Schatten norms

$$|\mathrm{Tr}\{AB\}| \leq \|A\|_1 \|B\|_\infty, \tag{3.59}$$

in Equation (3.58) together with the triangular and Cauchy-Schwarz inequality we obtain

$$\begin{aligned}&\left|\mathbb{E}_X\left(\mathrm{Tr}\left\{\mathbb{E}_{\boldsymbol{\Theta}_{R,j}}\left(B^{(2)}_{R,j}(\boldsymbol{\theta}_{R,j},x;\rho_0)\right)\right\}\left[V_j,\mathbb{E}_{\boldsymbol{\Theta}_{L,j}}\left(H(\boldsymbol{\theta}_{L,j},x)\right)\right]^{\otimes 2}\right)\right| \leq \\ &\mathbb{E}_X\left(\left|\mathrm{Tr}\left\{\mathbb{E}_{\boldsymbol{\Theta}_{R,j}}\left(B^{(2)}_{R,j}(\boldsymbol{\theta}_{R,j},x;\rho_0)\right)\right\}\left[V_j,\mathbb{E}_{\boldsymbol{\Theta}_{L,j}}\left(H(\boldsymbol{\theta}_{L,j},x)\right)\right]^{\otimes 2}\right|\right) \leq \\ &\mathbb{E}_X\left(\left\|\mathbb{E}_{\boldsymbol{\Theta}_{L,j}}\left(\left[V_j,H(\boldsymbol{\theta}_{L,j},x)\right]^{\otimes 2}\right)\right\|_\infty \left\|\mathbb{E}_{\boldsymbol{\Theta}_{R,j}}\left(B_{R,j}(\boldsymbol{\theta}_{R,j},x;\rho_0)\right)\right\|_1\right).\end{aligned} \tag{3.60}$$

Substituting in the previous equation the property [52]

$$\left\|\left[V_j,H(\boldsymbol{\theta}_{L,j},x)\right]^{\otimes 2}\right\|_\infty \leq 4\|V_j\|^2_\infty\|H\|^2_\infty, \tag{3.61}$$

and defining

$$\mathcal{B}^{(2)}_{R,j}(\rho_0) = \mathbb{E}_X\left(\left\|\mathbb{E}_{\boldsymbol{\Theta}_{R,j}}\left(B\right)^{(2)}_{R,j}\left(\boldsymbol{\theta}_{R,j},x;\rho_0\right)\right\|_1\right), \tag{3.62}$$

we obtain

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\Theta}_{R,j}}\left(\mathbb{E}_{\boldsymbol{\Theta}_{L,j}}\left(\mathbb{E}_X\left(\mathrm{Tr}\left\{B^{(2)}_{R,j}(\boldsymbol{\theta}_{R,j},x;\rho_0)\left[V_j,H(\boldsymbol{\theta}_{L,j},x)\right]^{\otimes 2}\right\}\right)\right)\right) &\leq \\ 4\|V_j\|^2_\infty\|H\|^2_\infty\,\mathcal{B}^{(2)}_{R,j}(\rho_0).&\end{aligned} \tag{3.63}$$

Following the same steps for Equation (3.57), we can bound this quantity as

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{\Theta}_{R,j}}\left(\mathbb{E}_{\boldsymbol{\Theta}_{L,j}}\left(\mathbb{E}_X\left(\mathrm{Tr}\left\{B^{(2)}_{L,j}(\boldsymbol{\theta}_{L,j},x;H)[V_j,\rho_R(\boldsymbol{\theta}_{R,j},0)]^{\otimes 2}\right\}\right)\right)\right) \\ &\qquad\leq 4\|V_j\|^2_\infty\|\rho_0\|^2_\infty\,\mathcal{B}^{(2)}_{L,j}(H),\end{aligned} \tag{3.64}$$

where we defined analogously

$$\mathcal{B}^{(2)}_{L,j}(H) = \mathbb{E}_X\left(\left\|\mathbb{E}_{\boldsymbol{\Theta}_{L,j}}\left(B^{(2)}_{L,j}(\boldsymbol{\theta}_{L,j},x;H)\right)\right\|_1\right). \tag{3.65}$$

Notice that we could interchange the x-dependency in Equation (3.57) and Equation (3.56) with no effect in the final bounds. The reason is that Equation (3.61) eliminates the x-dependency in the term where it is applied. We can compact the results from Equations (3.63) and (3.64) using the triangular inequality in

$$\left|\mathrm{Var}_{\boldsymbol{\Theta}}\left(\partial_j h_{\boldsymbol{\theta}}(0)\right) - \mathrm{Var}_{\boldsymbol{\Theta}}\left(\mathbb{E}_X\left(\partial_j h_{\boldsymbol{\theta}}(x)\right)\right)\right| \leq \\ 4\|V_j\|_\infty^2 \left(\|H\|_\infty^2\ \mathcal{B}^{(2)}_{R,j}(\rho_0) + \ \|\rho_0\|_\infty^2\ \mathcal{B}^{(2)}_{L,j}(h)\right) \tag{3.66}$$

□

3.6.2. Proof of Lemma 3.1

We start by bounding the right absorption witness of the $(l+1)$-th layer with the triangular and Hölder's inequality as
$\mathcal{B}^{(2)}_{R,l+1}(\rho_0) = \mathbb{E}_X\left(\left\|\mathbb{E}_{\boldsymbol{\Theta}_{l+1}}\left(u(\boldsymbol{\theta}_{l+1})^{\otimes 2} V(x)^{\otimes 2}\ \mathbb{E}_{\boldsymbol{\Theta}_{R,l}}\left(\rho_l(\boldsymbol{\theta}_{R,l}, x)\right) V^\dagger(x)^{\otimes 2} u^\dagger(\boldsymbol{\theta}_{l+1})^{\otimes 2}\right)\right\|_2\right) \leq$
$\mathbb{E}_X\left(\left\|\mathbb{E}_{\boldsymbol{\Theta}_{l+1}}\left(u(\boldsymbol{\theta}_{l+1})^{\otimes 2} V(x)^{\otimes 2}\ \mathbb{E}_{\boldsymbol{\Theta}_{R,l}}\left(B^{(2)}_{R,l}(\boldsymbol{\theta}_{R,l}, x; \rho_0)\right) V^\dagger(x)^{\otimes 2} u^\dagger(\boldsymbol{\theta}_{l+1})^{\otimes 2}\right)\right\|_1\right) +$
$\mathbb{E}_X\left(\left\|\mathbb{E}_{\boldsymbol{\Theta}_{l+1}}\left(u(\boldsymbol{\theta}_{l+1})^{\otimes 2} V(x)^{\otimes 2} - u(\boldsymbol{\theta}_{l+1})^{\otimes 2}\right)\right\|_1 \left\|\mathbb{E}_{\boldsymbol{\Theta}_{R,l}}\left(\rho_l(\boldsymbol{\theta}_{R,l}, 0)\right)\right\|_\infty\right).$

The second term can be identified as the layerwise absorption witness from Definition 3.2. The first term of the equation above can be bounded as
$\mathbb{E}_X\left(\left\|\mathbb{E}_{\boldsymbol{\Theta}_{l+1}}\left(u(\boldsymbol{\theta}_{l+1})^{\otimes 2} V(x)^{\otimes 2}\ \mathbb{E}_{\boldsymbol{\Theta}_{R,l}}\left(B^{(2)}_{R,l}(\boldsymbol{\theta}_{R,l}, x; \rho_0)\right) V^\dagger(x)^{\otimes 2} u^\dagger(\boldsymbol{\theta}_{l+1})^{\otimes 2}\right)\right\|_1\right) \leq$
$\mathbb{E}_X\left(\mathbb{E}_{\boldsymbol{\Theta}_{l+1}}\left(\left\|u(\boldsymbol{\theta}_{l+1})^{\otimes 2} V(x)^{\otimes 2}\ \mathbb{E}_{\boldsymbol{\Theta}_{R,l}}\left(B^{(2)}_{R,l}(\boldsymbol{\theta}_{R,l}, x; \rho_0)\right) V^\dagger(x)^{\otimes 2} u^\dagger(\boldsymbol{\theta}_{l+1})^{\otimes 2}\right\|_1\right)\right) =$
$\mathcal{B}^{(2)}_{R,l}(\rho_0)$.

Arranging both results together we can find

$$\mathcal{B}^{(2)}_{R,l+1}(\rho_0) \leq \mathcal{B}^{(2)}_{R,l}(\rho_0) + \|\rho_0\|_\infty^2 \mathcal{A}^{(2)}_{l+1}. \tag{3.67}$$

Equivalently for the left part of the circuit, and counting layers backward we find

$$\mathcal{B}^{(2)}_{L,l}(H) \leq \mathcal{B}^{(2)}_{L,l+1}(\rho_0) + \|H\|_\infty^2 \mathcal{A}^{(2)}_{l}. \tag{3.68}$$

□

3.6.3. Details on harmonic representation of QRU models

As stated in Equation (3.21), the wavefunction after a re-uploading circuit and before measurement can be expressed as

$$|\psi(x)\rangle = \sum_{j=1}^{2^n} \sum_{k=-K}^{K} c_{k,j} e^{ikx} |j\rangle. \tag{3.69}$$

The coefficients $c_{k,j}$ form the matrix $\boldsymbol{C} \in \mathbb{C}^{2^n \times (2K+1)}$. Each column (indexed with k) represents the corresponding term e^{ikx}. The states $|j\rangle$ are elements of any basis of choice. Each row (indexed with j) corresponds to the x-dependent amplitude attached $|j\rangle$. The quantity $p_j(x) = \langle j|\psi(x)\rangle$ is a trigonometric polynomial, $p_j(x) = \sum_{k=-K}^{K} c_{k,j} e^{ikx}$. Such polynomial can be represented as a vector $p_j = \{c_{k,j}\}_{-K \leq k \leq +K}$. In this vector representation, the multiplication of polynomials corresponds to convolution as

$$p(x)q(x) = \sum_{k=-K_p}^{K_p} p_k e^{ikx} \sum_{k=-K_q}^{K_q} q_k e^{ikx} = (p * q)(x), \tag{3.70}$$

We consider the three operations that can be applied to the harmonic representation.

Parameterized gates Applying a parameterized gate on the quantum state maps into applying the unitary representation of that gate to each column individually, the same way it would be done in state vector simulation for each (fixed) k.

Data-encoding gates Adding a data-encoding gate involves convolution, when $\boldsymbol{C}$ is expressed in the eigenbasis of the data generator. Each row j corresponds to the j-th (integer) eigenvalue λ_j from the spectrum of the data-encoding generator, $\mathcal{K}_g$. The vector representation of polynomials p_j is convoluted with the vector $e_{\lambda_j} = [\delta_{k=\lambda_j}]_{-K \leq k \leq +K}$.

Measurement For the measurement, we express $\boldsymbol{C}$ the basis of the observable. Secondly the representation the transpose conjugate of the wavefunction is computed from $\boldsymbol{C}$, by taking its conjugate and reverting the rows indexing $c_{j,k} = c_{j,-k}$. Finally the hypothesis function $h_{\boldsymbol{\theta}}(x)$ can be obtained as a linear combination of the rows of the result of the convolution weighted by the corresponding eigenvalue of the measurement operator.

3.6.4. Proof of Theorem 3.2

We assume now that the parameterized gates between two consecutive encoding steps are random unitaries sampled from the Haar measure of the group $\mathcal{SU}(N)$. Notice that data-independent operations leave the norm of coefficients associated with the same frequency invariant. Random unitaries output random states for any input. Random states give rise to a probability distribution sampled from a uniform Dirichlet [107], also known as Porter-Thomas [116] distribution. Under this assumption, no basis has any preference over any other, and the application of the data encoding layer *transports* as many coefficients as dictated by the spectrum $\mathcal{K}_{g_j}$ to the corresponding new frequencies. Notice that it is irrelevant which coefficients are transported, and also they are randomly chosen. The weights for the elements of each frequency are then described by a Dirichlet distribution with parameters given by the convoluted spectrum. Formally,

$$\sum_j |c_{k,j}|^2 \sim \mathrm{Dir}\left(\left(\mathcal{K}_{g_1} * \ldots * \mathcal{K}_{g_j} * \ldots * \mathcal{K}_{g_L}\right)(k)\right), \tag{3.71}$$

where $*$ denotes the convolution operator. Notice that the index k runs over all non-zero entries of the convoluted spectra. ☐

3.6.5. Dirichlet distribution

Definition 3.5 (Dirichlet distribution [107])
The Dirichlet distribution $\boldsymbol{x} \sim \mathrm{Dir}(\boldsymbol{\alpha})$ *parameterized by* $\alpha \in \mathbb{R}^N_{>0}$ *is supported on the* $(N-1)$*-standard simplex, i.e.,* $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$, $\|\boldsymbol{x}\|_1 = 1$. *It has the following probability density function with respect to the Lebesgue measure on* $\mathbb{R}^{N-1}$*:*

$$f_{\mathrm{Dir}}(\boldsymbol{x}, \boldsymbol{\alpha}) = \frac{\Gamma(\|\boldsymbol{\alpha}\|_1)}{\prod_{i=1}^N \Gamma(\alpha_i)} \prod_{i=1}^N x_i^{\alpha_i - 1}. \tag{3.72}$$

In this definition, $\Gamma(\cdot)$ is defined as

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt, \tag{3.73}$$

being the complex extension of the factorial for positive integers

$$\Gamma(n) = (n-1)!. \tag{3.74}$$

The Dirichlet distribution admits straightforward analytical calculations for the statistical moments of arbitrary order $\boldsymbol{k} = (k_1, k_2, \ldots, k_N)$,

$$\mathbb{E}_{\boldsymbol{x}\sim\mathrm{Dir}(\boldsymbol{\alpha})}\left(\prod_{i=1}^{N} x_i^{k_i}\right) = \frac{\Gamma(\|\boldsymbol{\alpha}\|_1)}{\Gamma(\|\boldsymbol{\alpha}\|_1 + \|\boldsymbol{k}\|_1)} \prod_{i=1}^{N} \frac{\Gamma(\alpha_i + k_i)}{\Gamma(\alpha_i)}. \tag{3.75}$$

In particular

$$\mathbb{E}_{\boldsymbol{x}\sim\mathrm{Dir}(\boldsymbol{\alpha})}(x_i) = \frac{\alpha_i}{\|\boldsymbol{\alpha}\|_1} \tag{3.76}$$

$$\mathrm{Var}_{\boldsymbol{x}\sim\mathrm{Dir}(\boldsymbol{\alpha})}(x_i) = \frac{\alpha_i\left(1 - \frac{\alpha_i}{\|\boldsymbol{\alpha}\|_1}\right)}{\|\boldsymbol{\alpha}\|_1(\|\boldsymbol{\alpha}\|_1 + 1)} \tag{3.77}$$

$$\mathrm{Cov}_{\boldsymbol{x}\sim\mathrm{Dir}(\boldsymbol{\alpha})}(x_i, x_j) = \frac{-\alpha_i\alpha_j}{\|\boldsymbol{\alpha}\|_1^2(\|\boldsymbol{\alpha}\|_1 + 1)} \tag{3.78}$$

3.6.6. Proof of Corollary 3.3

Let us express the wavefunction as the $\boldsymbol{C}$ matrix, such that the wavefunction is reconstructed from its elements as

$$|\psi(x)\rangle = \sum_{j=1}^{2^n} \sum_{k=-K}^{K} c_{k,j} e^{ik\mu x} |j\rangle, \tag{3.79}$$

where $|j\rangle$ is expressed in this case the eigenbasis of the observable of interest H. We are interested in the function

$$h(x) = \langle\psi(x)| H |\psi(x)\rangle, \tag{3.80}$$

which in the eigenbasis of H is

$$h(x) = \sum_j \sum_k \sum_l \lambda_j c_{k,j} c_{l,j}^* e^{i\mu(k-l)\ x}. \tag{3.81}$$

We give a bound now on the terms sharing the same frequencies

$$\left|\sum_j \sum_{k-l=\omega} \lambda_j c_{k,j} c_{l,j}^*\right|^2 \leq \|H\|_\lambda^2 \left|\sum_j \sum_{k-l=\omega} c_{k,j} c_{l,j}^*\right|^2 \leq$$
$$\|H\|_\lambda^2 \sum_{k-l=\omega} \left(\sum_j |c_{k,j}|^2\right)\left(\sum_j |c_{l,j}|^2\right), \tag{3.82}$$

where we used the triangular inequality and the Cauchy-Schwarz inequality. The last term is related to Theorem 3.2. Each of these elements is drawn from the Dirichlet distribution imposed by the spectrum $\mathcal{K}_g^{*L}$. The aggregation property of Dirichlet distributions allows us to directly work with the spectrums. The spectrum of interest is a modified convolution of $\mathcal{K}_g^{*L}$ with itself under an inversion of the variable, namely

$$\sum_{k-l=\omega} \mathcal{K}_g^{*L}(k)\mathcal{K}_g^{*L}(l) = \sum_k \mathcal{K}_g^{*L}(k)\mathcal{K}_g^{*L}(k-\omega) = \sum_k \mathcal{K}_g^{*L}(k)\mathcal{K}_g^{*L}(-(k-\omega)) = \left(\mathcal{K}_g^{*L} * \mathcal{K}_g'^{\,*L}\right)(\omega), \quad (3.83)$$

with $\mathcal{K}_g^{*L}(x) = \mathcal{K}_g'^{\,*L}(-x)$. In the case of symmetric spectra, both functions are equivalent. Recalling the properties of Dirichlet distributions, we can bound

$$\|H\|_\lambda^{-2} \left|a_\omega(\boldsymbol{\theta})\right|^2 \leq p_\omega \sim \mathrm{Dir}(\mathcal{K}_g^{*2L}(\omega)), \quad (3.84)$$

with

$$a_\omega(\boldsymbol{\theta}) = \sum_j \sum_{k-l=\omega} \lambda_j c_{k,j} c_{l,j}^* \quad (3.85)$$

$\square$

3.6.7. Proof of Theorem 3.3

We use tools from statistics to compute upper and lower bounds to the Lipschitz constant of a hypothesis function $\Lambda(h_{\boldsymbol{\theta}})$. We first recall the definition

$$\Lambda(h_{\boldsymbol{\theta}}) := \sum_{k=-K}^{K} \mu|k||a_k|, \quad (3.86)$$

and we recall the result from Corollary 3.3 in the limit of many re-uploadings. We know that

$$\mathbb{E}_{\boldsymbol{\Theta}}\left(\Lambda(h_{\boldsymbol{\theta}})\right) = \sum_{k=-K}^{K} \mu|k|\mathbb{E}_{\boldsymbol{\Theta}}\left(|a_k|\right). \quad (3.87)$$

We can use the known bound for the probability distribution underlying $|a_k|$, $|a_k|^2 \leq p_k \sim \mathrm{Dir}(\mathcal{K}_g^{*L})$. In particular, the marginals of the Dirichlet distribution are beta distributions [117]. The beta probability distribution with parameters α and β is defined as

$$\mathrm{Beta}_{\alpha,\beta}(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{(\alpha-1)}(1-x)^{(\beta-1)}. \tag{3.88}$$

In our case, see Equation (3.30), α is given by the Gaussian spectrum and $\beta = 1$. The expectation value of each element is given by

$$\mathbb{E}_{\boldsymbol{\Theta}}\left(|a_k|\right) \leq \int_0^1 dx \frac{\Gamma(\alpha_k+1)}{\Gamma(\alpha_k)} x^{(\alpha_k - 1/2)} = \frac{\alpha_k}{\alpha_k + 1/2} \leq 2\alpha_k, \tag{3.89}$$

using the property of the gamma function $\Gamma(1+x) = x\Gamma(x)$. The last inequality allows us to compute an upper bound in the limit of many re-uploadings by just computing

$$\begin{aligned}\sum_{k=-K}^{K} \mu k \mathbb{E}_{\boldsymbol{\Theta}}\left(|a_k|\right) \leq 2\|H\|_\lambda \sum_{k=-K}^{K} \mu k \mathcal{K}_g^{*2L}(k) \approx \\ 4\|H\|_\lambda \int_0^\infty \frac{\mu k}{\sqrt{4\pi L}\sigma_g} \exp\left(-\frac{k^2}{4\sigma_g^2 L}\right) = \\ \|H\|_\lambda \frac{4}{\sqrt{\pi}} \sigma_g \mu \sqrt{L},\end{aligned} \tag{3.90}$$

leading to the first result of the theorem.

The lower bound is easy to obtain by recalling the property $\|a\|_1 \geq \|a\|_2$. In our context, and in the limit of Gaussian processes

$$\mathbb{E}_{\boldsymbol{\Theta}}\left(\Lambda(h_{\boldsymbol{\theta}})\right)^2 \geq \|H\|_\lambda^2 \sum_{k=-K}^{K} \mu^2 k^2 \mathbb{E}_{\boldsymbol{\Theta}}\left(|a_k|^2\right) \approx \|H\|_\lambda^2 2L\mu^2\sigma_g^2, \tag{3.91}$$

leading to the second result of the theorem: $\mathbb{E}_{\boldsymbol{\Theta}}\left(\Lambda(h_{\boldsymbol{\theta}})\right) \geq \|H\|_\lambda \sigma_g \mu \sqrt{2L}$. ☐

3.6.8. Proof of Theorem 3.4

We are interested in knowing the probability of $\Lambda(h_{\boldsymbol{\theta}})$ to be larger than a certain reference value by some distance. We take this reference value to be the average $\mathbb{E}_{\boldsymbol{\Theta}}\left(\Lambda(h_{\boldsymbol{\theta}})\right)$, as in many statistics results. Consider now the lower-bound on the expectation value from Theorem 3.3. Since

$$\Lambda(h_{\boldsymbol{\theta}}) - \mathbb{E}_{\boldsymbol{\Theta}}\left(\Lambda(h_{\boldsymbol{\theta}})\right) \geq t \implies \Lambda(h_{\boldsymbol{\theta}}) - \|H\|_\lambda \mu\sqrt{2L}\sigma_g \geq t, \tag{3.92}$$

but not in the opposite direction, then

$$\mathrm{Prob}_{\mathbf{\Theta}}\left(\Lambda(h_{\boldsymbol{\theta}}) - \mathbb{E}_{\mathbf{\Theta}}\left(\Lambda(h_{\boldsymbol{\theta}})\right) \geq t\right) \leq \mathrm{Prob}_{\mathbf{\Theta}}\left(\Lambda(h_{\boldsymbol{\theta}}) - \|H\|_{\lambda}\mu\sqrt{2L}\sigma_g \geq t\right). \tag{3.93}$$

We can bound the right hand by considering Hoeffding's inequality [118]. Let X_i be a set of independent random variables, and let $X_i \in [a_i, b_i]$ *almost surely*, then

$$\mathrm{Prob}\left(\sum_i X_i - \mathbb{E}\left(\sum_i X_i\right) \geq t\right) \leq \frac{1}{2}\exp\left(\frac{-t^2}{\sum_i (a_i - b_i)^2}\right). \tag{3.94}$$

Hoeffding's inequality cannot be directly applied to a Dirichlet distribution since the variables are not independent. However, this problem can be overcome for this particular case by recalling the following property. If $X_i \sim \mathrm{Dir}(\alpha_i)$, then

$$X_i \sim \frac{Y_i}{V}, \tag{3.95}$$

with

$$Y_i \sim \mathrm{Gamma}(\alpha_i, \theta) \tag{3.96}$$

$$V = \sum_i Y_i \sim \mathrm{Gamma}\left(\sum_i \alpha_i, \theta\right) \tag{3.97}$$

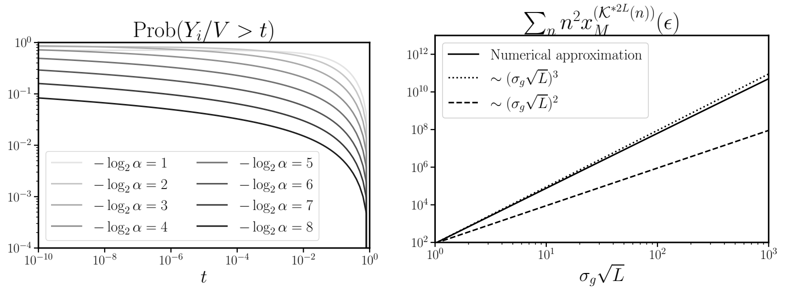
By changing the description of the Dirichlet distribution to the quotient of gamma distributions we can now apply Hoeffding's inequality. Without loss of generality, we can assume that all probabilities are bounded between 0 and 1, thus we can find a first bound by recalling

$$\sum_n n^2 \in \mathcal{O}((\|g\|_{\lambda} L)^3), \tag{3.98}$$

and thus

$$\begin{aligned}&\mathrm{Prob}_{\mathbf{\Theta}}\left(\Lambda(h_{\boldsymbol{\theta}}) - \mathbb{E}_{\mathbf{\Theta}}\left(\Lambda(h_{\boldsymbol{\theta}})\right) \geq t\right) \leq \\ &\quad \mathrm{Prob}_{\mathbf{\Theta}}\left(\Lambda(h_{\boldsymbol{\theta}}) - \|H\|_{\lambda}\sqrt{2L}\mu\sigma_g \geq t\right) \in \mathcal{O}\left(\exp\left(\frac{-t^2}{(\|g\|_{\lambda} L)^3}\right)\right)\end{aligned} \tag{3.99}$$

□



(a) Numerical calculations for Equation (3.101) for decreasing values of α. The values of the random variable concentrate in small values for t as α decays.

(b) Numerical approximation to Equation (3.103) for increasing $\sigma_g\sqrt{L}$. The value ε is set in this calculations to 10^{-10}

Figure 3.8.: Numerical auxiliary calculations for Equation (3.101) and Equation (3.103). These results substitute the non-accessible analytical treatment of the probability distributions of interest to obtain the bound given in Equation (3.36)

A tighter numerical bound

This bound can be however easily improved by recalling subgaussianity properties of the Gamma distribution. A random variable X is subgaussian if its cumulative distribution function decays faster than exponentially

$$\text{Prob}\left(|X| \geq t\right) \in \mathcal{O}\left(\exp\left(-t^2\right)\right), \tag{3.100}$$

for some positive constant K. We can compute this cumulative probability for the quotient of Gamma distributions as

$$\begin{aligned}\text{Prob}\left(\frac{Y_i}{V} \geq t\right) &= \int_0^\infty dx \int_x^{x/t} dy \frac{x^{\alpha-1}e^{-x}e^{-y}}{\Gamma(\alpha)} \\ &= \frac{1}{2^\alpha} - \frac{1}{(1+t^{-1})^\alpha} \in \mathcal{O}\left(\exp\left(-t^2\right)\right). \end{aligned}\tag{3.101}$$

These functions take the value 1 for $t = 0$ and decay until vanishing for $t = 1$. The decay is faster as $\alpha \to 0$, as it can be seen in Figure 3.8(a). We can thus recover Hoeffding's inequality with the observation that each X_i is bounded by the function in Equation (3.101). In particular, the variable

X_i is, with probability $1-\varepsilon$, smaller than

$$x_M^{(\alpha_i)}(\varepsilon) = \left(\left(\frac{1}{2^{\alpha_i}} - \varepsilon\right)^{-1/\alpha_i} - 1\right)^{-1}. \tag{3.102}$$

For a sufficiently small ε, the denominator of the exponent of Hoeffding's inequality becomes

$$\sum_n (a_n - b_n)^2 = 2 \sum_{n=1}^{\|g\|_2 L} n^2 x_M^{(\mathcal{K}_g^{*2L}(n))}(\varepsilon), \tag{3.103}$$

with $\mathcal{K}_g^{*2L}$ a Gaussian spectrum in the limit of large R. The Gaussian limit forces the intuition that only a small number of elements will contribute effectively, while for large values of n the corresponding Dirichlet variable is always so small that it has negligible influence in the Lipschitz constant. The description of the variable bounds in Equation (3.102) and the sum in Equation (3.103) prevent a straightforward analysis in terms of the relevant quantity $\sigma_g\sqrt{R}$. We can however make a numerical analysis, depicted in Figure 3.8(b). This calculation shows that the sum in Equation (3.103) follows a polynomial trend in $\sigma_g\sqrt{R}$, which is the variance of the resulting Gaussian spectrum. Therefore, we can update our previous version of Hoeffding's inequality to

$$\operatorname{Prob}_{\boldsymbol{\Theta}}\left(\Lambda(h_{\boldsymbol{\theta}}) - \|H\|_\lambda \sqrt{2L}\sigma_g \geq t\right) \in \mathcal{O}\left(\exp\left(\frac{-t^2}{(\sigma_g\sqrt{L})^3}\right)\right). \tag{3.104}$$

3.6.9. Extension to non-harmonic spectrum

The non-harmonic extension leads to an average of the elements in the trigonometric polynomial given by

$$\mathbb{E}_{\boldsymbol{\Theta}}\left(|a_{\vec{k}}|^2\right) = \frac{1}{\sqrt{(4\pi L)^D |\Sigma|}} \exp\left(-\frac{\vec{k}^T \Sigma^{-1} \vec{k}}{4L}\right), \tag{3.105}$$

where $|\Sigma|$ is the determinant of Σ. We compute now $\Lambda(h_{\boldsymbol{\theta}})$ following the steps from Section 3.6.7, given by

$$\mathbb{E}_{\boldsymbol{\Theta}}\left(\Lambda(h_{\boldsymbol{\theta}})\right) = \sum_{\vec{k}} |\vec{\mu}\cdot\vec{k}|\mathbb{E}_{\boldsymbol{\Theta}}\left(|a_{\vec{k}}|\right) \leq$$

$$\frac{2}{\sqrt{(4\pi L)^D|\Sigma|}}\int_{\mathbb{R}^D} d\vec{k}|\vec{\mu}\cdot\vec{k}|\exp\left(-\frac{\vec{k}^T\Sigma^{-1}\vec{k}}{4L}\right). \quad (3.106)$$

Notice that $d\vec{k}$ integrates over D-dimensional space. We perform now a change the variables to diagonalize $\Sigma = U^\dagger S U$, and consequently choose $\vec{l} = U\vec{k}$. The diagonal elements of S are denoted $\{s_j^2\}_j$ The quantity of interest is now $\vec{\mu}\cdot(U^\dagger\vec{l}) = (U\vec{\mu})\cdot\vec{l}$. Since U is unitary $d\vec{l} = d\vec{k}$

$$\mathbb{E}_{\boldsymbol{\Theta}}\left(\Lambda(h_{\boldsymbol{\theta}})\right) \leq \frac{2}{\sqrt{(4\pi L)^D|\Sigma|}}\int_{\mathbb{R}^D} d\vec{l}|(U\vec{\mu})\cdot\vec{l}|\exp\left(-\frac{\vec{l}^T S^{-1}\vec{l}}{4L}\right) \leq \quad (3.107)$$

$$\frac{2}{\sqrt{(4\pi L)^D|\Sigma|}}\sum_{j=1}^{D}|(U\vec{\mu})_j|\int_{\mathbb{R}^D} d\vec{l}|l_j|\exp\left(-\frac{\vec{l}^T S^{-1}\vec{l}}{4L}\right) \quad (3.108)$$

We focus now on the integral.

$$\int_{\mathbb{R}^D} d\vec{l}|l_j|\exp\left(-\frac{\vec{l}^T S^{-1}\vec{l}}{4L}\right) = \quad (3.109)$$

$$\int_{\mathbb{R}} dl_j|l_j|\exp\left(-\frac{\vec{l}_j^2}{4Ls_j^2}\right)\prod_{i\neq j}\int_{\mathbb{R}} dl_i\exp\left(-\frac{\vec{l}_i^2}{4Ls_i^2}\right) = \quad (3.110)$$

$$4Ls_j^2\prod_{i\neq j}\sqrt{4\pi L s_i^2} = \frac{1}{\pi}\sqrt{(4\pi L)^{D+1}}\sqrt{|\Sigma|}s_j \quad (3.111)$$

Plugging this result into Equation (3.108) we obtain

$$\mathbb{E}_{\boldsymbol{\Theta}}\left(\Lambda(h_{\boldsymbol{\theta}})\right) \leq \frac{(2\sqrt{L\pi})^{D+1}}{\pi}\frac{\sqrt{|\Sigma|}|\vec{\mu}U^\dagger|\cdot\sqrt{\vec{S}}}{\sqrt{(4\pi L)^D|\Sigma|}} = \frac{4\sqrt{L}}{\sqrt{\pi}}\sum_{j=1}^{D}|U\vec{\mu}|_j s_j \quad (3.112)$$

By means of Cauchy-Schwarz inequality, we can give a looser yet more comprehensive bound as

$$\mathbb{E}_{\boldsymbol{\Theta}}\left(\Lambda(h_{\boldsymbol{\theta}})\right) \leq \frac{4}{\sqrt{\pi}} \|\vec{\mu}\|_2 \sqrt{\mathrm{Tr}(\Sigma)} \sqrt{L}. \tag{3.113}$$

For the lower bound we follow Section 3.6.7 to obtain

$$\mathbb{E}_{\boldsymbol{\Theta}}\left(\Lambda(h_{\boldsymbol{\theta}})\right)^2 \geq \|H\|_\lambda^2 \sum_{k=-K}^{K} (\vec{\mu} \cdot \vec{k})^2 \mathbb{E}_{\boldsymbol{\Theta}}\left(|a_k|^2\right). \tag{3.114}$$

We recall the property [119, 120]

$$\int d^D \vec{k} f(\vec{k}) \exp\left(-\frac{\vec{k}^T \Sigma^{-1} k}{2}\right) = \sqrt{(2\pi)^D |\Sigma|} \exp\left(\frac{\vec{\nabla}^T \Sigma^{-1} \vec{\nabla}}{2}\right) f(\vec{k})\Bigg|_{\vec{k}=0}, \tag{3.115}$$

where $\vec{\nabla}_j = \partial / \partial \vec{k}_j$. Since $f(\vec{k}) = (\vec{\mu} \cdot \vec{k})^2$, we can reduce

$$\exp\left(\frac{\vec{\nabla}^T \Sigma^{-1} \vec{\nabla}}{2}\right) f(\vec{k})\Bigg|_{\vec{k}=0} = \vec{\mu}^T \Sigma \vec{\mu} \geq \|\vec{\mu}\|_2^2 \mathrm{min}_\Lambda(\Sigma), \tag{3.116}$$

yielding a result

$$\Lambda^2(h_{\boldsymbol{\theta}}) \geq \|H\|_\lambda^2 2L \|\mu\|_2^2 \mathrm{min}_\lambda(\Sigma). \tag{3.117}$$

$\square$

A simple example

We illustrate the spectral convolution with an example. Consider a data generator whose spectrum and multiplicities are

$$\lambda = \{-\sqrt{2}-1, -\sqrt{2}, -1, 0, +1, +\sqrt{2}, +\sqrt{2}+1\} \tag{3.118}$$

$$m(\lambda) = \{1, 1, 1, 2, 1, 1, 1\} \tag{3.119}$$

Any frequency resulting from the L-fold application of such data generator can be written as $\lambda_{k,l} = k\sqrt{2} + l$ where $-L \leq k, l \leq L$ are integers. The corresponding frequency content can therefore be represented as a two-dimensional tensor A. The elements of A follow a 2-dimensional Dirichlet distribution, in the sense of Theorem 3.2, given by the convoluted kernel

$$\mathcal{K}_g^{*L} = \frac{1}{8}\begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix}^{*L} \tag{3.120}$$

In the limit of large L, the central limit theorem applies exactly in the same way as in the harmonic case, and the L-fold convolution tends towards a multivariate Gaussian kernel with $[0, 0]$ mean and covariance matrix

$$\Sigma = \frac{L}{2}\begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}. \tag{3.121}$$

CHAPTER 4

Parameterized quantum circuits as universal generative models for continuous multivariate distributions

4.1. Introduction

Parameterized quantum circuits are the centerpiece of numerous approaches to machine learning on quantum computers, motivated by several near-term hardware limitations [4, 78]. The use of these models for supervised learning has been widely studied, for example in solving regression problems [121] where the goal is to assign continuous labels to data points. Variational algorithms have also been explored in so-called generative modelling tasks, where the objective is to generate new samples following a distribution that generated the training data [122].

A prominent example is the Quantum Boltzmann Machine (QBM) as introduced in [123], modelling distributions with discrete support, in which data is represented as the thermal state of an Ising model. In [124] the training based on the relative entropy of QBM for more general models is investigated, and it is shown that the training cannot be simulated with classical computers unless BPP=BQP. The QBM has been compared

---

The contents of this chapter have been published in [36].

in-extenso in [125] with another quantum generative model for discrete variables, the Quantum Circuit Born Machines (QCBM). In QCBM [45] a probability distribution over n-bit strings is stored in a n-qubit pure state. They may be trained to minimize the maximum mean discrepancy but also as generators in Generative Adversarial Networks (GANs) settings [93, 95, 126, 127], yielding the so-called Qu-GAN.

Going beyond distributions with discrete support, an approach to model distributions where the random variable can in principle take any value within a continuous interval has been introduced in [128]. In such models, the quantum circuit takes classical randomness as input and outputs expectation values, consequently, we call this model *expectation value samplers*. This model has been used as a quantum generator in the context of quantum generative adversarial networks (GAN), with applications including image generation [129, 130], high energy physics [131] and chemistry [132]. In all these applications the training is performed in the GAN setting.

While for QCBMs, the expressivity and universality have been clarified [44, 45], in contrast, expectation value sampling models are not so well understood. In particular, an interesting feature of expectation value sampling is that the dimension of the output is not inherently tied to the number of qubits used. In this chapter, we focus on the expressivity of the generators based on expectation value sampling depending on the number of qubits and the observable norm. In light of the numerous works [129–137] that successfully used expectation value samplers, this chapter aims to solidify the theoretical ground that supports them, by formally analyzing their expressivity. The formal analysis of the trainability of such models, while being a matter of critical importance, is not in the scope of this chapter.

The focus of this chapter is to show that parameterized quantum circuits are universal generative models and precisely characterize their expressivity. Our first main result is the existence of two universal families of expectation value samplers. We probe tight bounds relating to resource limitations, that is, necessary conditions on resources for expectation value sampling models to be universal. Specifically, we show that reaching universality for very high-dimensional distribution requires either a very large number of qubits or a very large number of measurements. This may serve as a backbone for future fine-grained resource cost analyses. As additional tools to analyze expressivity, we discuss choices of random variables, circuit encoding and observables, which we hope will guide future designs. Finally, we motivate the use of expectation value samplers with respect to other existing quantum generative models by providing a natural sampling task for these models.

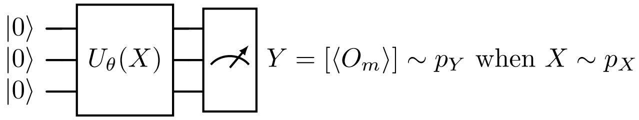


Figure 4.1.: Expectation value sampling model: a random vector is classically sampled. It is used to generate a random quantum state using a parameterized quantum circuit. The expectation values of fixed observables are returned as the output random vector.

4.2. Results

In this section, we introduce expectation value sampling models and define precisely the universality property for generative models. We state formally our first main result, namely the existence of two universal families of expectation value samplers. We then present our second main result, as the minimum resource requirements for expectation value samplers to be universal.

4.2.1. Expectation value sampling

The expectation value sampling procedure goes as follows. A random variable is classically sampled and used as input to a parameterized quantum circuit which specifies a random quantum state. The expectation values of fixed observables are measured and returned as another random variable. We illustrate this procedure in Figure 4.1, and provide a formal definition below.

Definition 4.1 (Expectation value sampling model)
An expectation value sampling model on n qubits is defined by $(U_\theta, \mathbf{O}, p_X)$, where $U_\theta : \mathcal{X} \subseteq \mathbb{R}^L \to \mathcal{U}(2^n)$ is a parameterized quantum circuit taking data as input and returning a matrix in the 2^n-dimensional unitary group $\mathcal{U}(2^n)$, $\mathbf{O} = (O_m)_{1\leq m \leq M}$ is a vector of M observables, and $p_X : \mathcal{X} \subseteq \mathbb{R}^L \to \mathbb{R}$ is the input distribution. We define the associated mapping f as follows:

$$x \in \mathcal{X} \xrightarrow{f} \left(\langle 0 | \, U_\theta(x)^\dagger O_m U_\theta(x) \, | 0 \rangle \right)_{1\leq m \leq M}. \tag{4.1}$$

The output of the model is a sample drawn from the distribution p_Y with $Y = f(X) \sim p_Y$ when $X \sim p_X$.

It is important to note that, in contrast to the quantum circuit Born machine, the randomness does not come from the measurement process, but from the classical randomness provided as an input to the quantum circuit. Another difference is that expectation value samplers have continuous support (absolutely continuous random variables, see Supplementary Information). The central question of this chapter is whether such a model can generate any multivariate distribution, more precisely whether expectation value sampling is a universal generative model, according to the definition we will give in the next subsection.

4.2.2. Universal generative model family

In this subsection, we precisely define universal generative model families. We choose the Wasserstein distance to quantify the closeness of two distributions. For reasons we detail in the Supplementary Information, While the Wasserstein distance is impractical for training purposes, it is still a relevant metric in the analysis of the expressivity. A main reason is because it naturally arises as the mathematical concepts we use in this chapter are related to optimal transport. Other common losses are the Kullback-Leibler Divergence, which is not well suited to this chapter as it is not symmetric and not a true metric. Another frequently used loss is the Maximum Mean Discrepancy which is easier to compute, but relative to a chosen kernel, and therefore not an absolute property of the closeness between two distribution.

Definition 4.2 (Universal generative model family)
A generative model is a family of parameterized sampling procedures which enable the sampling from a corresponding set of M-dimensional probability density functions $\mathcal{P}(\mathcal{X})$ on $\mathcal{X} \subseteq \mathbb{R}^M$.

A generative model is called universal if for every probability density function q on $\mathcal{X}$ there exists a sequence $\{p_k | p_k \in \mathcal{P}(\mathcal{X})\}_{1\leq k \leq \infty}$ such that it converges to q in the Wasserstein distance W.

$$W(q, p_k) \xrightarrow[k\to\infty]{} 0 \tag{4.2}$$

Importantly, universality is defined on a given support noted as $\mathcal{X}$ in Definition 4.2. For this chapter, we choose the support as the hypercube $\mathcal{X} = [-1, 1]^M$, because the first step of most machine learning pipelines is to rescale the data to fit on a given interval, usually $[-1, 1]$. Notably, choosing universality on a cubic support $[-1, 1]^M$ allows the expression of fully independent variables. Restricting $\mathcal{X}$ to smaller subsets would yield constraints on the dependence relationships expressible. For example,

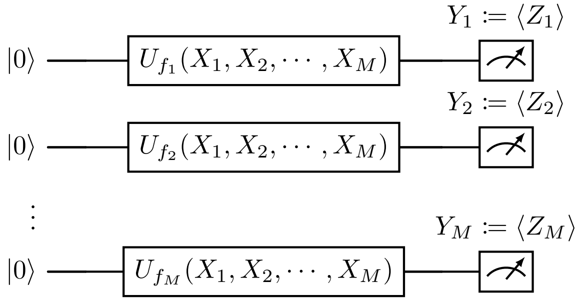

$Y_1 := \langle Z_1 \rangle$
$|0\rangle$
$U_{f_1}(X_1, X_2, \cdots, X_M)$
$Y_2 := \langle Z_2 \rangle$
$|0\rangle$
$U_{f_2}(X_1, X_2, \cdots, X_M)$
$\vdots$
$Y_M := \langle Z_M \rangle$
$|0\rangle$
$U_{f_M}(X_1, X_2, \cdots, X_M)$

Figure 4.2.: *Product Encoding* Circuit as a universal generator yielding the random variable $Y = g(X)$ when $X \sim U([0,1]^M)$, by stacking circuits from [25] U_f approximating f, and defining $f_m := \sqrt{(g_m+1)/2}$.

proving the universality of a family of models for Dirichlet distributions would correspond to choosing $\mathcal{X}$ in the hyperplane where all variables sum up to one and are positive.

4.2.3. Two families of Expectation Value Samplers as Universal Generators

First, building on work by [25], we show that n-qubit expectation value samplers with constant observable norm are universal for M-dimensional distributions with constant support radius, for $M = n$.

Theorem 4.1
For any M, for all M-dimensional probability density functions p_Z with support included in $[-1,1]^M$, and for all accuracy $\varepsilon > 0$ there exists a M-qubit circuit U and set of M observables $\mathbf{O}$ with unit spectral norm $\|O_m\| = 1$ such that the expectation value sampling model $(U, \mathbf{O}, p_X)$ where p_X is the uniform distribution on $[0,1]^M$ yields a probability density function p_Y that is ε-close to p_Z in the Wasserstein distance.

We derive an explicit construction, illustrated in Figure 4.2, for this circuit, which yields product states, hence we name it "*product encoding*". It uses the same number of qubits as the dimension of the output distribution. This construction defeats one advantage of expectation value samplers, that is the dimension of the output not being directly linked to the number of qubits used. This raises the question of the existence of a more qubit-frugal family of universal circuits, in which the output dimension is (much) larger

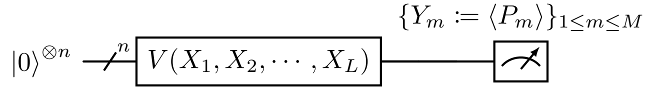

$\{Y_m := \langle P_m \rangle\}_{1 \leq m \leq M}$
$|0\rangle^{\otimes n}$
n
$V(X_1, X_2, \cdots, X_L)$

Figure 4.3.: *Observable Dense Encoding* Circuit as a universal generator, based on a universal state preparation circuit V, with each parameterized gate replaced by a circuit from [25]. $n = \log(M+1)$ and $P_m = 2M \left|m\right\rangle \left\langle m\right| - I$.

than the qubit number. We show the existence of such a family, at the cost of allowing for observables to have large norms. We formalize this in the following theorem and illustrate the explicit construction of the circuit in figure Figure 4.3 and call it "*observable-dense encoding*" expectation value sampler.

Theorem 4.2
For any M, for all M-dimensional probability density functions p_Z with support included in $[-1,1]^M$, and for all accuracy $\varepsilon > 0$ there exists a $n = \Theta(\log M)$-qubit circuit U taking L input variables and set of M observables $\mathbf{O}$ with spectral norm $\|O_m\| \in \Theta(M)$ such that the expectation value sampling model $(U, \mathbf{O}, p_X)$ where p_X is the uniform distribution on $[0,1]^L$ yields a probability density p_Y that is ε-close to p_Z in the Wasserstein distance.

In this subsection, we have provided *sufficient* conditions for expectation value samplers to be universal. In particular, we have shown the existence of two extremal families of parameterized quantum circuits that are universal generators on $[-1,1]^M$. There is a *product encoding* design, illustrated in Figure 4.2, with $n = M$ qubits and with unit norm observables (local Pauli), and an *observable-dense encoding* design, illustrated in Figure 4.3 with $n = \log(M)$ qubits and M norm observables (amplified probabilities of bitstrings). While one has a large number of qubits for a constant observable norm, the other has a logarithmic number of qubits but large observable norms. This hints that there might be a trade-off between the number of measurements and the number of qubits necessary to achieve universality. In the next section, we prove this is the case, by proving some *necessary* universality conditions.

4.2.4. Necessary conditions for universality

In this subsection, we prove two necessary conditions on resources for expectation value samplers to be universal. As previously mentioned, an appealing feature of expectation value sampling models is that the dimension of the output vector is *a priori* independent of the number of qubits n, unlike in the case of quantum Born machines where each qubit corresponds to exactly one binary random variable. In particular, we can imagine using just a single qubit with an arbitrary number of observables O_m to generate an M-dimensional random vector. However, it is obvious that in this case, the random variables corresponding to each observable cannot all be fully independent. Indeed, any observable can be expressed as a linear combination of the three Pauli matrices. Therefore the distribution output by a single qubit expectation value sampler will have at most three degrees of independence, and for $M > 3$ it is impossible to reach universality because it is impossible to approximate e.g. the 4-dimensional uniform distribution. Extending this reasoning to several qubits, we find the first necessary condition, that the dimension of the target dimension has to be lower or equal to the dimension of the space of observables. We formalize this in the following lemma.

Lemma 4.1
For an n-qubit expectation value sampling model $(U_\theta(x), \mathbf{O}, p_X)$ to be able to approximate any distribution with support in $[-1,1]^M$ to any accuracy $\varepsilon > 0$, it is necessary that $M \leq 4^n - 1$.

The second necessary condition for an expectation value sampling model to be universal on distributions with support in $[-1,1]^M$ can be derived using a combination of Holevo's bound found in [138] and Chernoff bound. We formalize it in the following theorem.

Theorem 4.3
For an n-qubit expectation value sampling model $(U_\theta, \mathbf{O}, p_X)$ to be able to approximate any distribution with support in $[-1,1]^M$ to any accuracy $\varepsilon > 0$ with respect to the Wasserstein distance, it is necessary that for every $m \leq M$, both equations are satisfied:

$$\lambda_{\min}(O_m) \leq -1 + \varepsilon \text{ and } \lambda_{\max}(O_m) \geq +1 - \varepsilon\,, \tag{4.3}$$

$$n \in \Omega\left(\frac{M(1-\varepsilon)^2}{\|O_m\|^2}\right). \tag{4.4}$$

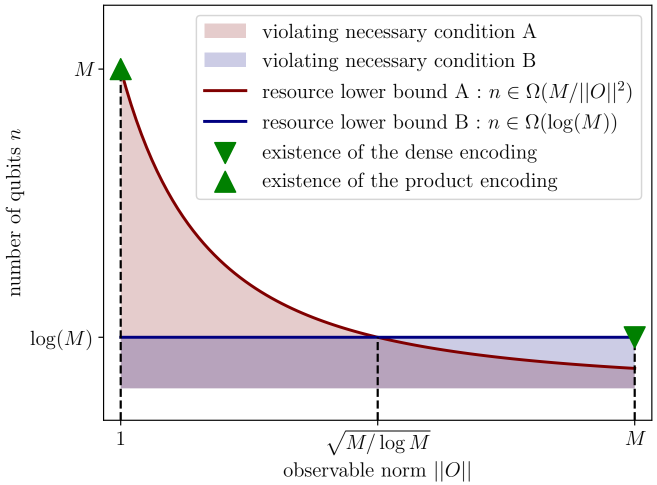


Figure 4.4.: Visual summary of results. We show the asymptotic use of resources for expectation value samplers to reach universality for a M-dimensional target distribution: the necessary conditions, as well as the existence of families as in Figure 4.2 and in Figure 4.3.

with $\lambda_{\min/\max}(O)$ *returning respectively the minimum and maximum eigenvalues of observable* O*.*

The combination of both previous necessary conditions is the second central result of this chapter. It formalizes that even if expectation value sampling models may output arbitrary large dimensional distributions, in practice their expressivity is limited by the number of qubits and observables. It is a natural question to ask whether the universal families we found previously saturate the above necessary conditions. We illustrate such considerations in Figure 4.4. For target distributions with exponentially large dimensionality, the two universal families are (almost) asymptotically optimal.

We may conjecture that there exists a family of (Pareto) optimal circuits, balancing between observable norms and qubit numbers, with varying *encoding densities* in between the two extremal ones. In practice, the observable spectral norm relates to the number of measurements required to approximate it up to an additive accuracy. This highlights a trade-off between the number of qubits and the number of measurements, or space versus time complexity, which we formalize in the next subsection.

4.2.5. Trade-off: qubits vs measurements

To estimate an expectation value up to a desired constant additive error, the number of measurements required is proportional to the norm of the observables. Therefore, large observable norms require a large number of measurements.

Lemma 4.2 (informal)
To guarantee that the M-dimensional distribution coming from an expectation value sampler performing T measurements is ε-close in the Wasserstein distance to the ideal shot-noise-free distribution, it is sufficient that

$$T \in \Theta\left(\frac{M\|O\|}{\varepsilon^2}\right). \tag{4.5}$$

Therefore the necessary conditions we derived previously ultimately highlight a trade-off between the number of qubits and the number of measurements. We show that to be universal for very high-dimensional distributions on $\mathcal{X}$, expectation value samplers need either a very large number of qubits or a very large number of measurements. This is in contrast with the initial intuition that expectation value samplers may generate high-dimensional distributions independently from the number of qubits.

4.3. Discussions

The arguments we present to derive resource lower bounds necessary for achieving universality in expectation value samplers (EVS) are driven by the number of independent variables of the target distribution. Because our definition of universality allows for full independence relationships across all dimensions, the number of independent variables is equal to the dimension of the target distribution. A more refined perspective emerges when considering families of distributions with inherent dependencies and allows for more economical sampling strategies than the ones requiring full independence as presented in Figure 4.4. Two concrete examples serve to emphasize this point. First, consider the family of distributions that are M-dimensional but have their support confined within a 2-dimensional linear subspace. In this case, using the expectation value sampler described in [25] with the observables $\sqrt{2}X, \sqrt{2}Z$, we achieve a universal generator on a single qubit for this 2-linear family, independent of the target distribution dimension M. This economy of representation underscores the capacity of EVS to exploit structural sparsity effectively. Next, consider M-dimensional Dirichlet distributions, characterized by non-negative coordinates summing to unity. Here, the observable dense encoding circuit, configured with $n = \log M$ qubits, enables universality with the raw probabilities as observables (Figure 4.3). This model achieves universal representation with exponentially fewer qubits compared to the general setting that allows for full independence, leveraging that the Born rule naturally gives rise to Dirichlet distributions. These findings suggest that EVS models are particularly well-suited for generating highly correlated distributions. The Dirichlet case is illustrative, as its normalization condition aligns seamlessly with that of quantum states, lending itself naturally to the expectation value sampling paradigm.

Building on this, we discuss the impact of design choices in EVS, specifically the selection of observables, the encoding $U(x)$, and the input distribution p_X, all of which are detailed further in the Supplementary Information. The choice of observables directly influences the subspace of the distribution spanned by the EVS. Furthermore, regarding the encoding structure $U(x)$, we realize that, analogously to the way parameterized quantum circuits may have an exact finite Fourier decomposition, expectation value samplers may have an exact finite Generalized Polynomial Chaos Expansion. In this context, the choice of the uniform distribution as an input distribution arises naturally, while in contrast to classical GANs, a Gaussian distribution yields undesirable inductive bias. Thanks to this proximity between the supervised and the generative context, results on the expressivity of quantum reuploading models [35] transfer naturally to the encoding circuits of expectation value samplers. In addition, as the Expectation Value samplers effectively learn a function in full analogy to how it is done in supervised learning, it is clear that much of the current discussion on the trainability of parameterized circuits for supervised learning, and of more general variational algorithms will apply to the Expectation Value samplers. It will have the same bottlenecks of overly expressive architectures [18], but also the various classes of new techniques that mitigate these by designs that balance trainability and dequantizability [21, 22, 139], and other more practical methods which mitigate trainability problems [59].

While we characterized important properties of expectation value samplers, we did not address the question of whether it is a good idea to use them. Expectation value samplers cannot be proven to be a path for certain types of quantum advantage as directly as is the case in Quantum Circuit Born Machines. QCBM are well suited for quantum use cases by design as they rely on the Born rule for generation of samples, and so correspond to distributions obtained by a measurement of a genuinely quantum state. In contrast, expectation value samplers rely on classical randomness and, rather than requiring sampling from the full distribution of quantum measurements, are defined around expectation values only. Thus the hardness of simulating distributions from expectation value samplers does not connect straightforwardly to any hardness-of-sampling results established in the domain of quantum supremacy results [44]. Nonetheless, expectation value samplers still consist of genuinely quantum computations, and arguments for non-simulatability can be made. Assuming BQP is not in BPP, there exists no polynomial time algorithm that takes a classical description of an arbitrary expectation value sampler A as an input and outputs a sample from a distribution that is epsilon close to that of the output of A. Take as an example the case where a hard-to-simulate circuit does not depend on input data. This yields a Dirac delta distribution for which there exist classical samplers to efficiently sample from it, however, such classical samplers cannot be easily found based on the classical description of the expectation value sampler. It may be possible to construct stronger advantage arguments where we find the existence of an expectation value sampler such that its output distribution cannot be sampled by any polynomial-time randomized Turing machine (subject to standard assumptions). This raises the question: do expectation value samplers also correspond to any natural quantum task?

A natural domain of application for EVS emerges in many-body physics, particularly in the study of disordered systems. We propose the example of spin glasses, which with interaction strengths $J_{i,j}$ taken at random, are governed by the Hamiltonian:

$$H(J) = \sum_{\langle i,j \rangle} J_{i,j} S_i S_j - M \sum_i S_i \,. \tag{4.6}$$

In this model, the interaction strengths $J_{i,j}$ are taken at random. Consider the quench dynamics of such a system, initialized as the ground state of the Hamiltonian without any interaction $H(0)$, that is the zero state. The interactions are instantly turned on for a fixed time and a set of interactions is taken at random. Then one may compute the expectation value of a set of observables of interest such as local spin, or local correlators, measured using several copies of the same realization of interaction strengths. This data is inherently quantum, and the trotterization of this time evolution constitutes a solution in the form of an expectation value sampler where the output can be made arbitrary close to the target distribution. We exemplify such a circuit for a chain geometry on four qubits in Figure 4.5. Using two registers with different evolution time, one could even extract two-time correlation function $C(t,t') = \sum_i \langle S_i(t) S_i(t') \rangle$, which in turn give precious information about the Edwards-Anderson parameter, an order parameter for spin glasses.

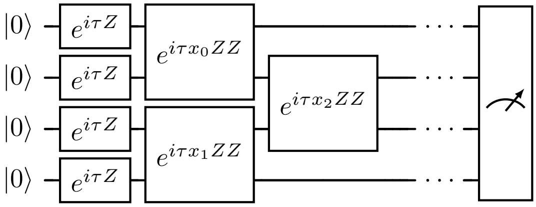


Figure 4.5.: Expectation value sampler as the trotterization of the spin glass Hamiltonian in Equation (4.6).

4.4. Methods

The proofs of all theorems are available in the Supplementary Information, but we provide high-level ideas in this section. First, we highlight a core concept in this chapter, variable transformation, and how we use it to prove universality. Subsequently, We explain the main steps of the proof and construction of the two universal families as a high-level summary of Section II of the SI .

4.4.1. Random Variable Transformation

A core concept in this chapter is that of random variable transformation. In this subsection, we introduce it and provide some of the associated fundamental properties.

The first step in the analysis of expectation value samplers is the observation that they are processes that map an input random vector (parameterizing the quantum circuit) to an output random vector (the expectation values of a set of observables). The literature on optimal transport and measure theory [140] states that for every pair of absolutely continuous random variables of the same dimension, there exists a mapping to transform one into the other.

Lemma 4.3
For every pair of absolutely continuous probability density functions $p_X \in \mathcal{P}(\mathcal{X} \subseteq \mathbb{R}^{\mathcal{M}})$ *and* $p_Y \in \mathcal{P}(\mathcal{Y} \subseteq \mathbb{R}^{\mathcal{M}})$*, there exists a mapping* $f : \mathcal{X} \to \mathcal{Y}$*, that maps* $Y \sim p_Y$ *to* $X \sim p_X$ *as* $Y = f(X)$*.*

We give intuition on how to construct this mapping to be bounded piece-wise continuous in the Supplementary information. Many generative modelling systems including Generative Adversarial Networks, Variational Auto-Encoders and normalizing flows rely on this idea to generate arbitrary distributions. In particular, the initial distribution is chosen to be simple, and then by altering the mapping applied to this random input, we obtain a rich spectrum of possible output distributions.

Then, sufficient conditions for a family of mappings to yield a universal generative model, in the sense of Definition 4.2, are well-known, and expressed in terms of the universality of mappings themselves. From [141], it is sufficient for a family of mappings to be dense in the set of all monotonically increasing functions in the pointwise convergence topology to yield a universal generative model. We explain the difference between pointwise topology and uniform topology in the Supplementary Information. Since these are sufficient conditions and monotonic functions are included in all functions, the following holds.

Lemma 4.4 ([141])
If a family of mappings $\mathcal{G} = \{g : \mathcal{X} \subseteq \mathbb{R}^M \to \mathcal{Y} \subseteq \mathbb{R}^M\}$ *is dense in the set of all functions in the pointwise convergence topology, then this family of mappings* $\mathcal{G}$ *together with a probability density function* p_X *with non zero support on* $\mathcal{X}$ *yields a universal generative model family on* $\mathcal{Y}$ *(cf Definition 4.2).*

In classical machine learning, such a notion for universality is common. It has been proven for several families of mappings in the context of normalizing flows, which are mappings with the additional property of being invertible: generic triangular mappings [140], neural networks mappings [142] and polynomial mappings [143].

In the next section, to prove the universality of expectation value samplers, we make explicit the connection between universal mapping families and the known results on the universality of parameterized quantum circuits as supervised learning models.

4.4.2. Universality proofs

In recent literature, several universality properties of parameterized quantum circuits have been proven. In [25], a family of single qubit quantum circuits with an increasing number of layers is proven to be universal in the uniform sense for continuous multidimensional input functions as complex coordinates of the quantum state in the computational basis.

We extend this existing result to fit our needs, as follows. First, we modify universality results from functions as coordinates in the computational basis to functions as the expectation value of an observable. Then we broaden the universality of quantum reuploading models to some discontinuous functions by relaxing the required strength of convergence. More precisely we go from the uniform density in bounded continuous functions to the pointwise density in bounded piece-wise continuous functions. Finally, by stacking M universal circuits, we extend universality to multivariate output functions. All these extensions of [25] together yield the theorem below.

Theorem 4.4
For every natural number M, for every mapping $f \in \mathcal{B}([0,1]^M \to [-1,1]^M)$, there exists a sequence of sets of M single qubit quantum circuits and unit norm observables (indexed by k).

$$\{(U_{k,m} : [0,1]^M \to \mathcal{U}(2), O_{k,m})_{1\leq m\leq M}\}_{1\leq k\leq\infty} \tag{4.7}$$

such that the sequence of functions $\{g_k\}_{1\leq k\leq\infty}$ defined as

$$g_{k,m}(x) = \langle 0| U_{k,m}(x)^\dagger O_{k,m} U_{k,m}(x) |0\rangle \tag{4.8}$$

converges pointwise to f, where $\mathcal{B}$ is the set of piecewise continuous functions and the norm of observable is the spectral norm.

The theorem above shows that there exists a family of M-qubit circuits with unit norm observables that yield a family of functions that is pointwise dense in the set of bounded piece-wise continuous functions. This matches the sufficient conditions of Lemma 4.4 for mappings to yield a universal generative model. This yields that there exists a family of expectation value sampling models as defined in Definition 4.1 and illustrated in Figure 4.2 that is universal in the sense of Definition 4.2.

For the *observable dense encoding* we follow a similar strategy, but instead, each output variable is encoded as the overlap between the output state and each computational basis state. With the normalisation of the quantum states the observables are projectors on computational basis states, amplified by a factor proportional to the dimension of the target distribution. More details are provided in the Supplementary Information.

4.5. Appendix

4.5.1. Universality Definition

In this appendix, we spend a bit more time justifying our choice for the definition of universality and relating it to concepts of interest. The definition of universality for generative models hinges on the notion of closeness between two random variables. In the case of generative modelling, *convergence in distribution*, a common concept in probability theory, is often sought. For example, the central limit theorem precisely states the average of any L independent random variables with mean μ and variance Σ converges in distribution to the normal distribution $\mathcal{N}(\mu, \Sigma/L)$.

Definition 4.3 (Universal generative model family)
A generative model is a family of parameterized sampling procedures which enable the sampling from a corresponding set of M-dimensional probability density functions $\mathcal{P}(\mathcal{X})$ on $\mathcal{X} \subseteq \mathbb{R}^M$.

A generative model is called universal if for every probability density function q on $\mathcal{X}$ there exists a sequence $\{p_k | p_k \in \mathcal{P}(\mathcal{X})\}_{1 \leq k \leq \infty}$ such that the sequence of random variables $X_k \sim p_k$ converges in distribution to $X \sim q$. Equivalently, this means that the sequence of cumulative distribution functions of p_k, which we call P_k, converges pointwise to the cumulative distribution function of q, which we call Q.

$$\forall x \in \mathcal{X}, \lim_{k \to \infty} P_k(x) = Q(x). \tag{4.9}$$

In this chapter, we will mostly consider distributions with finite support, because this is the case in most practical real-world problems. In particular, their probability density functions are integrable functions and convergence in distribution implies convergence in the first Wasserstein distance W_1 [144]. For completeness, we recall below the definition of the Wasserstein distance, also known as the Earth Mover's Distance, that we use in the context of this chapter.

Definition 4.4 (Wasserstein distance)
The k-th Wasserstein distance between two probability density functions p

and q on $[-1,1]^M$ *is defined as:*

$$W_k(p,q) = \left(\inf_{\gamma \in \Pi(p,q)} \int_{\mathbb{R}^2} \|x-y\|^k d\gamma(x,y) \right)^{\frac{1}{k}}, \tag{4.10}$$

where $\Pi(p,q)$ *is the set of couplings of* p *and* q*, and* $\|\cdot\|$ *denotes the Euclidean distance. The parameter* $k \geq 1$ *determines the so-called order of the Wasserstein distance.*

For the rest of the chapter, we will only consider the first-order Wasserstein distance ($k = 1$) and therefore simply refer to it as the Wasserstein distance.

Importantly, universality is defined on a given support noted as $\mathcal{X}$ in Definition 4.3. For this chapter, we choose the support to be the hypercube $\mathcal{X} = [-1,1]^M$, because the first step of most machine learning pipelines is to rescale the data to fit on a given interval. Note that the length of the hypercube can be rescaled, by rescaling the norm of the observables. Finally, since any distribution with infinite support but finite moments can be arbitrarily approximated by a distribution with finite support, this means that if we allow the observable norm to scale, we can also approximate any distribution with finite moments (but perhaps infinite support). We make this explicit below.

We consider a distribution with probability density function p_Y with infinite support $\mathbb{R}^M$ and finite moments. The goal is to find a sequence of random variables with bounded support probability density function that converges pointwise to p_y. We define the sequence of random variables $\{Y_k\}_{1 \leq k \leq \infty}$ whose probability density functions p_{Y_k} are proportional to that of Y on the hypercube $[-k-k_0, +k+k_0]^M$, where k_0 is the first integer such that Y has non-null support on the hypercube. This sequence converges in distribution to Y.

4.5.2. Universality Proofs

In this appendix, we provide the proofs for theorems 1 and 2 that state the universality of two families of EVS. First, we provide more details on random variable mapping which is the core concept of the proof in Section 4.5.2. For both proofs, we use Theorem 4.5 from [25]. For the first theorem, in Section 4.5.2 we modify universality results from functions as coordinates in the computational basis to functions as the expectation value of an observable. In Section 4.5.2 we broaden the universality of quantum reuploading models to some discontinuous functions by relaxing the required strength of convergence. More precisely we go from the uniform density in bounded continuous functions to the pointwise density in bounded piece-wise continuous functions. Finally, by stacking M universal circuits, we extend universality to multivariate output functions. All these extensions of [25] together yield Theorem 4. With this theorem, we show that the *product encoding* circuit satisfies the universal mapping property, and therefore, using the concept of random variable mapping, yields a universal generative model. For the *observable dense encoding* in Section 4.5.2 we follow a similar strategy, but instead, each output variable is encoded as the overlap between the output state and each computational basis state. With the normalisation of the quantum states the observables are projectors on computational basis states, amplified by a factor proportional to the dimension of the target distribution.

Constructive mapping between a random variable and the uniform distribution

Let us consider an absolutely continuous random variable Y with probability density function p_Y on a bounded set $[a, b]^M$. We recall a definition of an absolutely continuous variable below.

Definition 4.5
A random variable X is said to be absolutely continuous if its cumulative distribution function (CDF) can be expressed as the integral of a non-negative function, known as the probability density function (PDF).

This excludes for example Dirac deltas. In what follows we construct an invertible mapping to transform the uniform random variable X into this random variable $Y = [Y_k]$. We call $G_1 : [0, 1] \to [a, b]$ the cumulative distribution function of the marginal of Y_1. It is invertible and we define F_1 as its inverse,

$$Y_1 = F_1(X_1), X_1 = G_1(Y_1). \tag{4.11}$$

Next we consider the marginal of Y_2 conditioned by Y_1, we define the cumulative distribution $G_2 : [a, b]^2 \to [0, 1]$,

$$G_2(y_1, y_2) = P(Y_2 = y_2|Y_1 = y_1). \tag{4.12}$$

It is invertible with respect to Y_2,

$$G_2^{-1}(Y_1, X_2) = Y_2 \iff G_2(Y_1, Y_2) = X_2. \tag{4.13}$$

We define $F_2 : [0,1]^2 \to [a,b]$ as follows,

$$F_2(X_1, X_2) = G_2^{-1}(F_1(X_1), X_2). \tag{4.14}$$

We have $Y_2 = F_2(X_1, X_2)$. Continuing this process iteratively for all coordinates, it is possible to fully define the invertible mapping $F = [F_k]$ with inverse $G = [G_k]$ such that $Y = F(X)$. This is the essence of triangular mapping in [140]. In addition, this mapping is bounded and piece-wise continuous, because it is composed of inverse cumulative distribution functions of absolutely continuous variables on bounded support.

From state coordinate universality to expectation of observable universality

We start by recalling Theorem 4 from [25], which proves that the following circuit is universal. It has L layers, $\theta \in \mathbb{R}^{(M+2)\times L}$ parameters and for $x \in \mathbb{R}^M$ is defined as

$$U_\theta(x) \coloneqq \prod_{l=1}^{L} R_y(\theta_{0,l}) \left(\prod_{m=1}^{M} R_z(x_m \theta_{m,l}) \right) R_z(\theta_{M+1,l}). \tag{4.15}$$

Theorem 4.5 (from [25])
For any pair of functions and real number

$$(f \in \mathcal{C}([0,1]^M \to [0,1]), \phi \in \mathcal{C}([0,1]^M \to [0,2\pi)), \varepsilon > 0)$$

There exists a one qubit circuit $U : [0,1]^M \to \mathcal{U}(2)$ *s.t.*

$$\forall x, \left| \langle 1| U(x) |0\rangle - f(x) e^{i\phi(x)} \right| < \varepsilon. \tag{4.16}$$

In this chapter, $\mathcal{C}$ is the set of continuous functions. This theorem yields the universality of functions embedded in a quantum state in the uniform sense. In the context of expectation value sampling, we are interested in the universality of function as the expectation value of a unit norm observable, captured by the following theorem.

Theorem 4.6
For any function $g \in \mathcal{C}([0,1]^M \to [-1,1])$ *and for any* $\varepsilon > 0$*, there exists a one qubit circuit* $U(x) : [0,1]^M \to \mathcal{U}(2)$ *and an observable* O *with unit spectral norm* $\|O\| = 1$ *s.t.*

$$\forall x, \left| \langle 0| U^\dagger(x) O U(x) |0\rangle - g(x) \right| < \varepsilon. \tag{4.17}$$

Proof: We are given an arbitrary function $g \in \mathcal{C}([0,1]^M \to [-1,1])$ and $\varepsilon > 0$. We define the function $f = \sqrt{\frac{g+1}{2}}$ which is well defined on $\mathcal{C}([0,1]^M \to [0,1])$. We apply Theorem 4.5 to $(f, \phi = 0, \varepsilon/4)$ and get a circuit U that yields a state close to

$$|x\rangle = \sqrt{1-f(x)^2}\,|0\rangle + f(x)\,|1\rangle\,. \tag{4.18}$$

The Z expectation value of the above state is

$$\langle x|\,Z\,|x\rangle = 2f(x)^2 - 1 = g(x). \tag{4.19}$$

We are now going to prove that the expectation value is close to the target function g. First, we note that the square function is 2-Lipschitz on $[0,1]$ and therefore for every pair of real numbers $(x,y) \in [0,1]^2$

$$|x-y| < \varepsilon \implies \left|x^2 - y^2\right| < 2\varepsilon. \tag{4.20}$$

We define $p_{0/1}$ the probabilities of measuring $U(x)\,|0\rangle$ in state $|0\rangle$ and $|1\rangle$ respectively. Recalling that

$$|\langle 1|\,U(x)\,|0\rangle - f(x)| < \varepsilon/4, \tag{4.21}$$

we can write

$$\left|p_1 - f(x)^2\right| < \varepsilon/2 \tag{4.22}$$

$$\begin{aligned} \left|p_0 - (1-f(x)^2)\right| &= \\ \left|1 - |\langle 1|\,U(x)\,|0\rangle|^2 - (1-f(x)^2)\right| &< \varepsilon/2. \end{aligned} \tag{4.23}$$

Finally,

$$|\langle Z\rangle - g(x)| \leq \left|p_1 - f(x)^2\right| + \left|p_0 - (1-f(x)^2)\right| < \varepsilon. \tag{4.24}$$

This yields the uniform density of quantum functions in the set of bounded continuous functions.

From uniform density on continuous functions to pointwise density on discontinuous functions

We first start by highlighting the difference between uniform convergence and pointwise convergence and illustrate it with an example.

Definition 4.6 (Pointwise convergence)
Let $\{f_k | f_k : \mathcal{X} \to \mathbb{R}\}_{1 \leq k \leq \infty}$ *be a sequence of functions, and let* $f : \mathcal{X} \to \mathbb{R}$ *be another function defined on the same domain* $\mathcal{X}$. *We say that the sequence* $\{f_k\}$ *converges pointwise to* f *if, for each* $x \in \mathcal{X}$, *the sequence of real numbers* $\{f_k(x)\}_{1 \leq k \leq \infty}$ *converges to* $f(x)$ *as* k *approaches infinity,*

$$\lim_{n \to \infty} f_k(x) = f(x) \quad \textit{for all } x \in \mathcal{X}.$$

Definition 4.7 (Uniform convergence)
Let $\{f_k | f_k : \mathcal{X} \to \mathbb{R}\}_{1 \leq k \leq \infty}$ *be a sequence of functions, and let* $f : \mathcal{X} \to \mathbb{R}$ *be another function defined on the same domain* $\mathcal{X}$. *We say that the sequence* f_k *converges uniformly to* f *if, for any given* $\varepsilon > 0$, *there exists an* $K \in \mathbb{N}$ *such that for all* $k \geq K$ *and for all* $x \in \mathcal{X}$, *the difference* $|f_k(x) - f(x)|$ *is less than* ε,

$$\forall \varepsilon > 0, \exists K \in \mathbb{N} : \forall k \geq K, \forall x \in \mathcal{X}, |f_k(x) - f(x)| < \varepsilon.$$

Uniform convergence is stronger, it implies pointwise convergence, but the reverse is not true. For example, consider the step function f.

$$f(x) = \begin{cases} -1, & \text{if } x \in [-1, 0[\\ +1, & \text{if } x \in [0, +1] \end{cases} \tag{4.25}$$

It is impossible to define a sequence of continuous functions that would uniformly converge to it, however, it is possible to have a sequence of continuous functions that converges pointwise to it, see Figure 4.6.

$$f_k(x) = \begin{cases} -1, & \text{if } x \in [-1, 1/k[\\ 1 + kx, & \text{if } x \in [-1/k, 0] \\ +1, & \text{if } x \in]0, +1] \end{cases} \tag{4.26}$$

In [145] Baire defined hierarchical pointwise convergence classes of functions. Baire class 0 is the set of continuous functions, and the class c is the set of functions that are the pointwise limit of class $c - 1$. In particular in [146, 147], it was proven that bounded piece-wise continuous functions are of class 1. This means that for any bounded piece-wise continuous function f there exists a sequence of bounded continuous functions that converges pointwise to f. This means that bounded continuous functions are dense in bounded piece-wise continuous functions in the pointwise topology. Therefore, building on Theorem 4.6 we have the following theorem, writing $\mathcal{B}$ as the set of piecewise continuous functions.

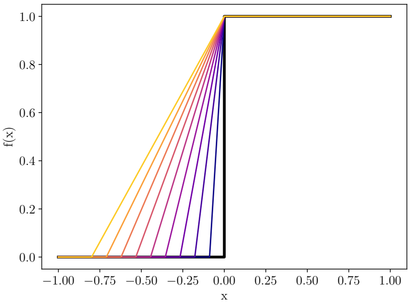


Figure 4.6.: A sequence of continuous functions converging pointwise but not uniformly to the step function.

Theorem 4.7
For any function $f : \mathcal{B}([0,1]^M \rightarrow [-1,1])$*, there exists a sequence (indexed by* k*) of one qubit circuit and observables with unit spectral norm,*

$$\{(U_k(x) : [0,1]^M \rightarrow \mathcal{U}(2), O_k)\}_{1\leq k \leq \infty} \tag{4.27}$$

such that the sequence of functions $\{g_k\}$ *with*

$$g_k(x) = \langle 0| U_k^\dagger(x) O_k U_k(x) |0\rangle \tag{4.28}$$

converges pointwise to f*.*

Construction of the observable dense encoding circuit

Note: We have used slight abuse of notations in the below proof to increase readability, specifically in the approximations noted as $\equiv_\varepsilon$.

Considering architecture such as in [148] and universal one qubit gate as $R_x(\alpha)R_z(\beta)R_x(\gamma)$, for any number of qubits, it is possible to design a circuit $U : [0, 2\pi)^L \rightarrow \mathcal{U}(2^n)$ made only of a finite number of fixed gates (CNOT and constant rotations) and parameterized σ_z rotations gates that can reach any pure quantum state when applied to state $|0\rangle$,

$$\forall \left|\psi\right\rangle, \exists \theta \in [0, 2\pi)^L, \left|\psi\right\rangle = U(\theta) \left|0\right\rangle. \tag{4.29}$$

Let's consider a distribution p_ψ over pure states. Because the architecture above can reach any pure state, there exists a corresponding distribution over the parameters p_θ such that the distribution $V(\theta) \left|0\right\rangle, \theta \sim p_\theta$ matches perfectly p_ψ in distribution. We define $g : [0,1]^L \to [0,2\pi)^L$ as the mapping that transforms the uniform distribution over $[0,1]^L$ into p_θ. We have $V(f(X))\left|0\right\rangle, X \sim U([0,1]^L)$ matches perfectly p_ψ in distribution, where V is composed of a finite number of fixed gates and L σ_z rotation gates parameterized by $g_l(X)$.

Lemma 4.5
For any distribution over pure states p_ψ, there exists a circuit architecture V made of constant gates and parameterized σ_z rotations, and a mapping g such that

$$X \sim U([0,1]^L), V(g(X))\left|0\right\rangle \sim p_\psi \tag{4.30}$$

Next, we decompose $R_z \circ g$ gates into a sequence of constant gates and parameterized σ_z rotations.

From [25], in the proof in the appendix, it is shown that there exists a quantum circuit W taking a multidimensional input and approximating the following parameterized quantum gate,

$$\begin{aligned}&\forall f : [0,1]^L \to [0,1], \phi : [0,1]^L \to [0,2\pi), \exists W,\\ &W(x) \equiv_\varepsilon \begin{bmatrix} \sqrt{1-f(x)^2}e^{+i\phi(x)} & -f(x)e^{+i\phi(x)} \\ f(x)e^{-i\phi(x)} & \sqrt{1-f(x)^2}e^{-i\phi(x)} \end{bmatrix}.\end{aligned} \tag{4.31}$$

Choosing $\phi = 0$, and $f = \sin g$, we have $\sqrt{1-f^2} = \cos g$, and we note that $W(x) = R_z \circ g$. We can conclude the following.

Lemma 4.6

$$\forall f : [0,1]^L \to [0,1], \exists W, R_z \circ f \equiv_\varepsilon W, \tag{4.32}$$

where W is a quantum circuit made of constant gates and R_z gates applied to individual components of x.

Combining both above lemmas we get the following.

Lemma 4.7
For any distribution over pure states p_ψ, there exists a circuit architecture V made of constant gates and parameterized z rotations such that

$$X \sim U([0,1]^L), V(X)\left|0\right\rangle \sim_\varepsilon p_\psi. \tag{4.33}$$

We are now going to use that lemma to prove that n-qubits expectation value samplers are universal for $\exp(n)$-dimensional distributions with constant support if the observables are allowed to have $\exp(n)$ norms.

We are given an arbitrary random variable Y following a M-dimensional distribution p_Y with support $[-1,1]^M$. We define the following state over n qubits with $M = 2^n - 1$.

$$|\psi(Y)\rangle = \sum_{m\leq M} Z_m |m\rangle + \sqrt{1 - \sum_{m\leq M} Z_m^2}\, |M+1\rangle \tag{4.34}$$

$$Z_m = \sqrt{\frac{Y_m + 1}{2M}} \tag{4.35}$$

We define as p_ψ as the probability density functions over states when $Y \sim p_Y$, using the previous lemma we get a circuit W composed only of constant gates and z rotations gates with one of the L parameters as input that approximates p_ψ. We define the observables $\forall m \leq M, O_m = 2M |m\rangle\langle m| - I$. They have spectral norm $\|O_m\| = 2M - 1 = \Theta(2^n)$. In addition, $\langle\psi(Y)| O_m |\psi(Y)\rangle = Y_m$. Therefore, the expectation value sampler $(W, O, U([0,1]^L))$ approximates p_Y. This concludes the proof to Theorem 2.

4.5.3. Proof of necessary resources for universality

In this appendix, we prove Theorem 3 about the necessary resources for EVS to achieve universality, which we recall below.

Theorem
For an n-qubit expectation value sampling model $(U_\theta, \mathbf{O}, p_X)$ to be able to approximate any distribution with support in $[-1,1]^M$ to any accuracy $\varepsilon > 0$ with respect to the Wasserstein distance, it is necessary that for every $m \leq M$:

1. $\lambda_{\min}(O_m) \leq -1 + \varepsilon$ *and* $\lambda_{\max}(O_m) \geq +1 - \varepsilon$
2. $n \in \Omega\left(\frac{M(1-\varepsilon)^2}{\Lambda(O_m)}\right) \subseteq \Omega\left(\frac{M(1-\varepsilon)^2}{\|O_m\|^2}\right)$

with $\lambda_{\min/\max}(O)$ returning respectively the minimum and maximum eigenvalues of observable O, and $\Lambda(O) \coloneqq -\lambda_{\min}(O)\lambda_{\max}(O)$.

Proof: Let's suppose there is an n-qubit expectation value scheme with M observables $\mathbf{O}$ that is able to approximate any distributions with support included in $[-1,1]^M$ to ε with respect to the first Wasserstein distance W_1.

Because the expectation value model is universal, it means that for any vertex of the hypercube $c \in \{-1,1\}^M$, it can approximate the Dirac delta at c. We then use the lemma below.

Lemma 4.8
Any distribution that is ε-close (in the Wasserstein distance) to the Dirac delta at a given point must have nonzero support within the Euclidean distance sphere centred in that point with radius ε.

This means that there is non-zero support on the ε sphere around c, and therefore there exists a quantum state ρ_c whose list of expectations is ε-close to that point. This yields the following result.

$$\forall c \in \{-1,1\}^M, \exists \rho_c, \sum_{m=1}^{M} (\mathrm{Tr}(O_m \rho_c) - c_m)^2 \leq \varepsilon^2. \tag{4.36}$$

Because the above is a sum of positive components, we can write $\forall m$

(a) if $c_m = 0$, then $-1-\varepsilon \leq \mathrm{Tr}(O_m \rho_c) \leq -1+\varepsilon$

(b) if $c_m = 1$, then $+1-\varepsilon \leq \mathrm{Tr}(O_m \rho_c) \leq +1+\varepsilon$

The above yields conditions on the spectrum of O_m. We note $\lambda_{\min}$ and $\lambda_{\max}$ respectively the minimum and maximum eigenvalues of O_m. We define $\gamma := 1-\varepsilon$. We know that $\forall \rho, \mathrm{Tr}(O\rho) \geq \lambda_{\min}$ therefore, $-\gamma \geq \lambda_{\min}$, the same reasoning applies for the maximum eigenvalue, yielding:

(a) $\lambda_{\min} \leq -\gamma$,

(b) $\lambda_{\max} \geq +\gamma$.

For the rest of the proof, we combine approaches from the proof of theorem 2.6 in [149] and that of B.1 in [150]. We define the two-outcome POVMs $\{E_m, I - E_m\}$ with

$$E_m = \frac{O_m - \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}} \tag{4.37}$$

We define $\beta := \frac{-\lambda_{\min}}{\lambda_{\max}-\lambda_{\min}}$. The spectral inequalities yield $\beta \geq \frac{1}{2}$. With this definition, the two conditions above translate to:

(a) $c_m = 0, p_0 := \mathrm{Tr}(E_m \rho_c) \leq \beta - \frac{\gamma}{\lambda_{\max}-\lambda_{\min}}$

(b) $c_m = 1, p_1 := \mathrm{Tr}(E_m \rho_c) \geq \beta + \frac{\gamma}{\lambda_{\max}-\lambda_{\min}}$

We define the amplified Positive Operator-Valued Measures (POVMs) that apply $\{E_m, I - E_m\}$ to $L \geq 1$ copies of ρ and return 1 if and only if at least βL copies of the original POVMs return 1.

From Holevo's bound (found as Theorem 5.1 in [138]), for the amplified scheme to correctly identify the corresponding bit $b_m = (c_m + 1)/2$ with probability q it is necessary that

$$nL \geq (1 - H(q))M, \tag{4.38}$$

where H is the binary entropy function.

We define the random variable $X_{m,l}$ which takes the value of the output of the POVM of the m-th observable on the l-th copy. We define $X_m^{(L)} := \frac{1}{L}\sum_l X_{m,l}$.

In the case $c_m = -1$, we have $\mathbb{E}[X_m^{(L)}] = p_0$. The probability of the amplified POVMs yielding the wrong output is

$$P(X_m^{(L)} > \beta) \leq P\left(X_m^{(L)} > p_0 + \frac{\gamma}{\lambda_{\max} - \lambda_{\min}}\right) \tag{4.39}$$

Recalling that $\beta \geq 1/2$, we can use the Chernoff bound on the Bernoulli variable $X_m^{(L)}$ and we get

$$P(X_m^{(L)} > p_0 + \frac{\gamma}{\lambda_{\max} - \lambda_{\min}}) \leq \exp - \frac{\gamma^2 L}{2(-\lambda_{\min})\lambda_{\max}} \tag{4.40}$$

We define $\Lambda = (-\lambda_{\min})\lambda_{\max}$, we have $\gamma^2 \leq \Lambda \leq \|O\|^2$.

For the probability of the amplified POVMs to yield the correct output with probability q it is necessary that $P(X_m^{(L)} \leq \beta) \geq q$.

Finally, we get

$$\log(1/q) \leq \frac{\gamma^2 L}{2\Lambda} \tag{4.41}$$

Combining Chernoff's and Holevo's inequalities, we conclude the proof of Theorem 3:

$$\Lambda \geq \frac{1 - H(q)}{\log(1/q)} \frac{\gamma^2 M}{n}. \tag{4.42}$$

Note: The above is a tighter condition than in Theorem 2.6 in [150] but falls back to it, when $\Lambda = \|O\|^2$, which corresponds to $\lambda_{\min} = -\|O\|$ and $\lambda_{\max} = \|O\|$. In the opposite scenario, we have $\Lambda = \gamma^2$, which corresponds to constant norm observables, yielding $n \in \Omega(M)$. The norm of observables affects the number of measurements to reach a desired additive accuracy.

Note: The above necessary conditions use results which involve a more general case where we assume that prior to measurement we use a general parameterized quantum channel which can also prepare mixed states.

However, we can reduce this to the special case of unitaries as well. By purification, mixed states can be mimicked by using $2n$ qubits pure states, since we are only interested in scalings, the factor 2 plays no role in the second condition of Theorem 3, and the necessary conditions also hold for pure states.

4.5.4. Approximation of expectation values

In this appendix, we justify the connection between the spectral norm of observables and the number of measurements. In practice, we do not have access to exact expectation values and we have to estimate them through sampling. This creates a distribution $p_{\hat{Y}}$ (n dimensional) slightly different from the distribution with exact expectation values p_Y. For simplicity, we assume that shot noise $\rho_\varepsilon \sim \mathcal{N}(0, \varepsilon I_n)$ (n dimensional) acts like an additional Gaussian noise with zero mean and standard deviation ε. ρ_ε and Y are independent and we consider the random variable $\hat{Y} = Y + \rho_\varepsilon$. Therefore the density $p_{\hat{Y}}$ is the convolution of p_Y and p_{ρ_ε}. Using Lemma 7.1.10 from [151], we know that the Wasserstein distance between p_Y and $p_{\hat{Y}}$, $W_p(p_Y, p_{\hat{Y}}) \in O(\varepsilon)$.

Lemma 4.9
An arbitrary expectation value sampler outputs an M-dimensional random vector $Y \sim p_Y$ with an infinite number of measurements, i.e. with access to exact expectation values. We consider the same circuit but with a finite number of measurements T that estimates expectation value by sampling and averaging for each observable and yields a random vector $\hat{Y} \sim p_{\hat{Y}}$. The number of measurements T required to guarantee that the Wasserstein distance between both distributions is smaller than ε satisfies

$$T \in \Theta\left(\frac{M\|O\|}{\varepsilon^2}\right). \tag{4.43}$$

In practice, different techniques exist to estimate expectation values of observables with different degrees of measurement efficiency, with shadow tomography techniques surpassing the “vanilla estimation”. For simplicity, we consider a vanilla estimation where each observable with norm $\|O\|$ is measured t times and the average is returned. This yields a shot noise close to the Gaussian model above with $\varepsilon^2 \in \Theta(\|O\|/t)$. The total number of measurements T is then $T = tM$, which yield $T \in \Theta(M\|O\|/\varepsilon^2)$

Note that the first result cannot be trivially applied to techniques such as shadow tomography. Indeed the corresponding shot noise ρ cannot in general be modeled by a Gaussian independent noise, at the least the covariance matrix will in general not be proportional to the identity.

4.5.5. Additional expressivity tools

In this appendix, we provide additional tools to analyze the expressivity of expectation value samplers. Specifically the choice of observabes, and the choice of random variable encoding.

Primary mapping and the choice of observables

We are using the standard Pauli basis for the space of $2^n \times 2^n$ Hermitian operators $\mathbf{P}$. It is composed of all possible combinations of n Pauli matrices $\sigma_{0,1,2,3}$, which yields $|\mathbf{P}| = 4^n$. We formalize as follows

$$\mathbf{P} := (P_k, k \in \{0,1,2,3\}^n) \tag{4.44}$$

$$= (\otimes_{1 \leq i \leq n} \sigma_{k_i}, k_i \in \{0,1,2,3\}). \tag{4.45}$$

Any vector of M observables $\mathbf{O} = (O_m)$, can be expressed as a linear mapping applied on the vector of all Pauli strings: $\mathbf{O} = A\mathbf{P}$, where A is an $M \times 4^n$ matrix, and $\mathbf{P}$ is a 4^n dimensional vector. Therefore the distribution associated with the Pauli basis encompasses any distributions, which leads us to define the primary mapping as follows.

Definition 4.8 (Primary mapping)
The primary mapping g of an n-qubit encoding circuit $U_\theta(x)$ is defined as the mapping of the associated expectation value sampling model with the Pauli basis $\mathbf{P}$, defined as $(U_\theta(x), \mathbf{P}, p_X)$ according to the definition in the main body. It can be expressed as follows

$$x \in [0, 2\pi)^N \xrightarrow{g} \left(\langle 0| U_\theta(x)^\dagger P_k U_\theta(x) |0\rangle\right)_{1 \leq k \leq 2^n}. \tag{4.46}$$

It yields the 4^n-dimensional random variable $Z = g(X)$ when $X \sim p_X$.

It is easy to see that any distribution obtained by an expectation value sampling model can always be expressed by considering an intermediary output of the 4^n set of observables, followed by the linear mapping A. We capture this idea in the following theorem.

Lemma 4.10
Given an encoding circuit $U_\theta(x)$ and random variable with distribution p_X, for any choice of observables $\mathbf{O}$, the expectation value sampling model

$(U_\theta(x), \mathbf{O}, p_X)$ is a linear transformation of the expectation value sampling model $(U_\theta(x), \mathbf{P}, p_X)$, where $\mathbf{P}$ is the Pauli basis per definition of the primary mapping in Definition 4.8.

This concept of primary mapping has immediate consequences on the possible correlation of output variables of expectation value sampling models. For a given data encoding part on n qubits, the primary mapping will yield a random variable Z with a covariance matrix C_z. Because any expectation value sampling model based on the same data encoding is a linear transformation of the primary mapping, the number of uncorrelated variables is limited by the number of non-null eigenvalues of the covariance matrix C_z, which is upper bounded in any case by $4^n - 1$.

Lemma 4.11
Given an encoding circuit $U(x)$ over n qubits and random variable with distribution p_X, we note L the number of non-zero eigenvalues of the covariance matrix of the primary mapping. For any choice of observables, any expectation value model using $(U(x), p_X)$ will yield at most L uncorrelated variables. In addition $L \leq 4^n - 1$.

Random variable encoding as a polynomial chaos expansion

After focusing on the choice of observables, in this subsection, we analyze the impact of the choice of the input random variable and the circuit encoding on the expressivity. We propose the polynomial chaos expansion [152] as a useful tool to analyze the expressivity of expectation value sampling models, as the analogue of the Fourier decomposition. The general polynomial chaos expansion is a representation of random variables as a vector in a Hilbert space of orthogonal functions, as defined below.

Definition 4.9 (Generalized Polynomial Chaos Expansion)
A generalized chaos expansion is characterized by a probability density function p_X defined on the support $\mathcal{X} \subseteq \mathbb{R}^M$ with finite moments (usually chosen as standard distributions, such as Gaussian or uniform). This choice defines an inner product for functions in $\{f : \mathcal{X} \to \mathbb{R}\}$:

$$\langle f|g\rangle_{p_X} := \int_{\mathcal{X}} f^*(x)g(x)p_X(x)dx. \tag{4.47}$$

This choice of inner product comes with the choice of an ordered family of functions, usually polynomials, that are orthonormal with respect to the above inner product.

$$\Phi_{p_X} = \{\phi_l : \mathcal{X} \to \mathbb{R}, \forall(k,l), |\phi_l\rangle \langle\phi_k| = \delta_{k,l}\} \tag{4.48}$$

A generalized chaos expansion is a representation of a random variable Y with probability density p_Y as a vector α in this Hilbert space, such that:

$$Y = \sum_{l=0}^{\infty} \alpha_l \phi_l(X) \sim p_Y, X \sim p_X \tag{4.49}$$

This provides a Hilbert space as a potential structure to study random variable mappings. In particular, one of the main results of polynomial chaos expansion is that they are universal generative models.

Common pairs of distribution and associated orthogonal polynomials family can be found in Table 4.1. In the context of expectation value sampling, we focus on the family of functions orthonormal with respect to the inner product associated with the uniform distribution on $\mathcal{X} = [0, 2\pi)^M$

$$\Phi = \{ \prod_{1 \le m \le M} e^{ik_m x_m}, k \in \mathbb{Z}^M \} \tag{4.50}$$

Quantum reuploading circuits are a widely used class of parameterized quantum circuits that output a function of the data. They are used in a regressive context, where optimization techniques are used such that their output fits a target function. It has been widely studied and used that if they use integer-valued spectrum Hamiltonian, their output hypothesis function can be decomposed as an exact finite Fourier series [26]. This means that there exists $c_{\mathbf{k},l} \in \mathbb{C}$ such that the hypothesis function f can be exactly written as a finite Fourier series:

$$f(x) = \sum_{k_0, k_1, \cdots, k_M = -K}^{+K} c_{\mathbf{k},l} \prod_{1 \le m \le M} e^{ik_m x_m}, . \tag{4.51}$$

This fact extends to a generative modelling context where expectation value sampling models using integer-valued quantum reuploading circuits yield distributions with an exact finite polynomial chaos expansion. We formalize this below.

Theorem 4.8
Any expectation value sampling model using a quantum reuploading model with integer-valued spectrum $U_\theta(x)$, together with the uniform distribution on $[0, 2\pi)^M$ outputs a random variable Y that has an exact finite polynomial chaos expansion for any choice of observables.

This subsection formalizes the tight connection between quantum circuits used in a regressive context with their use in a generative context. Therefore

Normal distribution	Hermite polynomials
Uniform distribution	Legendre polynomials
Exponential distribution	Laguerre polynomials
Beta distribution	Jacobi polynomials

Table 4.1.: Pairs of distributions and corresponding orthonormal families commonly used in General Polynomial Chaos expansion.

it is expected that, beyond the universality, many properties of such models can be transferred from a regressive context to a generative context, but we leave that for future work.

Choice of input distribution

Lastly, we discuss the choice of input random variable, noted as p_X. We highlight that the universality theorems in this chapter apply to the uniform distribution. This, together with generalized chaos expansion considerations makes the uniform distribution a natural choice for the input random variable of expectation value samplers. This is due mostly to the fact that it has a bounded support. Indeed, it is known that the most commonly used parameterized quantum circuits are periodic, in particular when they have a finite Fourier decomposition. This is also why the universality of quantum reuploading circuits is proven for functions on bounded domains, corresponding to a half period. In contrast, let us consider an expectation value sampler $(U, \mathbf{O}, p_X)$, where the input random variable follows a Gaussian distribution $X \sim \mathcal{N}(0, 1)$, which has unbounded support and for which the mapping f is 1-periodic. $X = 0$ is the highest probability event and will yield the same output as a very low probability event, for example, $X = 100$. This means that a very low probability input event and a very high probability input event will yield the same output sample, which is a feature rarely considered desirable. This choice of uniform distribution is in contrast to classical GANs where Gaussian distributions are typically preferred.

CHAPTER 5

Quantum Advantage in Learning Quantum Dynamics

5.1. Introduction

Identifying when quantum computers provide an advantage in learning tasks is a central challenge of quantum machine learning. Problems involving quantum systems are natural candidates, but many of their associated learning tasks can still be solved efficiently by classical computers. For instance, [153] shows that predicting ground state properties within a phase is classically tractable with limited training data. Nonetheless, the prevailing intuition is that access to a device that can efficiently simulate the target quantum system should be advantageous. While earlier works identified contrived problems exhibiting such separation [21, 22], identifying quantum-classical learning separations in natural settings remains a compelling open question.

We address this challenge by considering the supervised learning problem of *learning unknown Hamiltonian dynamics* from classical data. Specifically, we define a family of functions $f_\alpha(x) = \langle\psi(x,\alpha)|O|\psi(x,\alpha)\rangle$, where $|\psi(x,\alpha)\rangle = e^{iH(x,\alpha)t}|\psi_0\rangle$ is a time-evolved quantum state under a pa-

---

The contents of this chapter have been published in [37].

rameterized Hamiltonian $H(x, \alpha)$ for some fixed time t. The task is to learn f_α (in the PAC sense, which we explain shortly) from a dataset $\mathcal{D} = \{(x_j, f_\alpha(x_j))\}_j$ for some fixed, unknown α. Our settings assume a more restricted (and maybe more realistic) access to the dynamics than found in related works in the literature [154, 155], in particular excluding the use of Hamiltonian learning methods.

Our contributions can be summarized as follows. We introduce a quantum subroutine to extract the Fourier decomposition of PQC-based functions, which we call *Fourier coefficient extraction* subroutine. We then use this subroutine to define a quantum learning algorithm to solve the unknown Hamiltonian dynamics problem, which we formalize as the Hamiltonian dynamics concept class (defined shortly) in the probably approximately correct (PAC) framework. In our approach, we use the fact that the quantum dynamics class can be approximated via Hamiltonian simulation by a class of functions built around parameterized quantum circuits, which we call the PQC-based functions concept class. The latter class will serve as the effective hypothesis family of our learning algorithm. Our learning algorithm is then proven to be correct for both concept classes. The algorithm is efficient when the number of unknown parameters (of the PQC or Hamiltonian) scales logarithmically with the system size. We also prove that no classical algorithm can solve this learning task under complexity-theoretic assumptions, yielding a separation for this natural problem. We also consider the much more general setting of polynomially many unknown parameters. For this case, we analyze the potential and limitations of generalization of our method for this broader class of quantum dynamics problems. We identify conditional no-gos for provably efficient learners, but also propose a heuristic method for the problem, which may work in cases beyond what can be proven analytically.

This chapter is structured as follows. Section 5.2 introduces the *Fourier coefficient extraction* algorithm. Section 5.3 applies this to learn PQCs, and Section 5.4 extends this to Hamiltonian evolution. Section 5.6 discusses limitations and heuristic extensions. We finish this chapter with a brief conclusion section in Section 5.6.2.

5.1.1. Parameterized Quantum circuits

Parameterized quantum circuits, and in particular variational methods [4], have been at the centre of recent approaches to quantum machine learning and were extensively studied [156]. Here, we focus on parameterized quantum circuits where the inputs and other tunable parameters appear repeatedly as Pauli rotations.

Definition 5.1 (Pauli encoding)
A Pauli-encoded circuit is a parametrized quantum circuit on n qubits $U : \alpha \in [0,1]^d \to \mathcal{U}(2^n)$. It is composed of $N_f \in \mathrm{poly}(n)$ fixed unitary gates. There are d parameters, each reuploaded L times, such that there are $dL \in \mathrm{poly}(n)$ parametrized gates:

$$\{V_{s,j}(\alpha) \coloneqq e^{i\pi P_{s,j}\alpha_j}\}_{(s,j)\in\{1,\cdots,L\}\times\{1,\cdots,d\}}$$

where $P_{s,j}$ are Pauli strings.

It has been shown that such Pauli Quantum Circuits admit a finite Fourier representation [26].

Lemma 5.1
Any circuit U as defined in Definition 5.1 admits a finite Fourier representation with frequencies included in $\mathcal{L}' = l \in \{-L, \cdots, +L\}^d$ as follow:

$$\begin{aligned} |\phi(\alpha)\rangle &= U(\alpha)\,|0\rangle && (5.1)\\ &= \sum_{k\in\{0,1\}^n}\sum_{l\in\mathcal{L}'} a_{l,k}e^{i\pi\alpha\cdot l}\,|k\rangle\ . && (5.2) \end{aligned}$$

Measuring the expectation value of some observable O for such a state results in what we call a *PQC function*, an input-output mapping as follows,

$$f(\alpha) = \langle 0|\,U^\dagger(\alpha) O U(\alpha)\,|0\rangle\ . \tag{5.3}$$

In the case of quantum reuploading models with Pauli encodings as in Definition 5.1, we have that

$$f(\alpha) = \sum_{l\in\{-2L,\cdots,2L\}^d} b_l e^{i\pi\alpha\cdot l}\,. \tag{5.4}$$

We call the coefficients b_l the *Fourier coefficients* of the *PQC function* f.

In the next section, we provide a subroutine for sampling from them. Building on this subroutine, we provide a sampling-based algorithm for their estimation.

5.2. Fourier coefficient extraction algorithm

5.2.1. Fourier representation of parameterized circuits

For our quantum learning algorithm, we introduce a new subroutine that allows us to prepare a state that amplitude-encodes the coefficients b_l

of a PQC function f, and describe a sampling-based method for their extraction. First we observe that the amplitudes of the output state of a PQC function admit a finite Fourier representation, as in Lemma 5.2.

Lemma 5.2
Any circuit U as defined in Definition 2.4 admits a finite Fourier representation as follows:

$$|\phi(x)\rangle = U(x)|0\rangle = \sum_{k\in\{0,1\}^n} \sum_{l\in[-L,+L]^D} a_{l,k} e^{i\pi x\cdot l} |k\rangle \ . \tag{5.5}$$

As we explain in Section 5.7.5, it is possible to estimate the coefficients given just appropriate black-box access to the classical function f. However, here we assume access to the gate-decomposition of the circuit, which yields a more elegant and significantly more gate-frugal method. Based on this gate decomposition, we propose an algorithm that transforms the description of this circuit to the description of a circuit with an additional register for frequencies.

Theorem 5.1
There exists an algorithm $\mathcal{A}$ that given the description of any parameterized circuit U as defined in Definition 5.1, returns the description of a non-parameterized quantum circuit $U' = \mathcal{A}(U)$ on n' qubits with L' gates, such that it prepares the Fourier representation state of U as follows:

$$|\phi'\rangle = \mathcal{A}(U)|0\rangle = \sum_{k\in\{0,1\}^n} \sum_{l\in\mathcal{L}'} a_{l,k} |l\rangle |k\rangle \ . \tag{5.6}$$

with $n' = n + d\lceil\log(2L+1)\rceil + 1$ qubits and $L' = N_f + L(2n + d\lceil\log(2L+1)\rceil)$ gates.

The detailed algorithm and the proof for this theorem are provided in the Section 5.7.3. We provide a high-level explanation of this algorithm. The fixed gates (the ones without data uploading) are left unchanged by the algorithm $\mathcal{A}$. For the reuploading gates, we note that any data Pauli-uploading gates can be transformed into a Z string by adding appropriate basis change gates. The algorithm $\mathcal{A}$ transforms a data-encoding gate with a Z Pauli on the bitstring x into an increment (decrement) gate on the frequency register controlled on the even (odd) parity of x. This is illustrated in Figure 5.1.

5.2.2. Fourier representation of expectation values

As seen in Lemma 5.1, when the inputs α are Pauli-encoded, the PQC function has a finite Fourier decomposition. We can connect Theorem 5.1 with this representation to obtain a sampling algorithm for the coefficients b_l as given in Equation (5.4). First, we construct a circuit that returns the Fourier decomposition of such a quantum function amplitude encoded on a quantum state.

The key to do so is to realise that as $f(\alpha) = \langle 0| U(\alpha)^\dagger P U(\alpha) |0\rangle$, applying the algorithm $\mathcal{A}$ to the state $U(\alpha)^\dagger P U(\alpha) |0\rangle$ and then post-selecting the all-zero state yields the Fourier decomposition of the quantum function f as a quantum state. The full proof is in the Section 5.7.3.

Corollary 5.1
For every circuit U as defined in Definition 5.1 and Pauli observable P, we define the quantum function f:

$$\alpha \in \mathbb{R}^d \xrightarrow{f} \langle 0| U(\alpha)^\dagger P U(\alpha) |0\rangle \ . \tag{5.7}$$

f has a finite Fourier representation, there exists a vector $b \in \mathbb{C}^{d(4L+1)}$ indexed by $l \in \mathcal{L} \coloneqq \{-2L, \cdots, +2L\}^d$ such that

$$f(\alpha) = \sum_{l \in \mathcal{L}} b_l e^{i\pi\alpha\cdot l} \, . \tag{5.8}$$

Then there is a quantum algorithm $\mathcal{A}$ with complexity $O(\mathrm{poly}(n, d))$ that retrieves the state amplitude encoding of the Fourier coefficients of f on the $|0\rangle$ subspace.

$$\mathcal{A}(U) |0\rangle |0\rangle = \frac{1}{\|b\|_2} \sum_{l \in \mathcal{L}} b_l |l\rangle |0\rangle + |\cdots\rangle \left|0^\perp\right\rangle \ . \tag{5.9}$$

Note that the probabilistic component here is unavoidable, as in general there is no reason for the function f to have a Fourier decomposition which is a unit vector. Note that the theorem above is specific to observables that are Pauli strings P. In the Section 5.7.3, we prove that this result can efficiently be extended to a broader range of observables O, such as linear combinations of Pauli terms with a polynomial number of terms and polynomial spectral norm and local projectors.

Using $\mathcal{A}$ as a subroutine, we can perform *Fourier coefficient extraction*, i.e., extract any coefficient b_l up to additive error, by using controlled versions of $\mathcal{A}(U)$ in a Hadamard test.

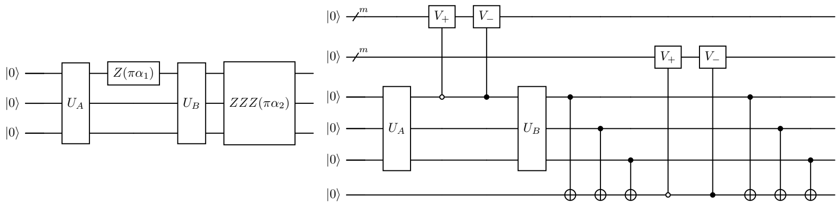


Figure 5.1.: (left) A parameterized circuit $U(\alpha)$ yielding a state $\sum_k \sum_l a_{l,k} e^{i\pi l\cdot\alpha} |k\rangle$. (right) The corresponding circuit $\mathcal{A}(U)$ returning a state amplitude-encoding Fourier coefficients as $\sum_k \sum_l a_{l,k} |l\rangle |k\rangle$
. .

Corollary 5.2
With the same premise as in Corollary 5.1, there exists a $\mathrm{poly}(n, \varepsilon^{-1})$ *quantum algorithm that retrieves* b_l *up to additive error* ε *for every* l*.*

Furthermore, the Fourier amplitude-encoded state realizes a particular quantum feature map, which can be used to construct kernel-based machine learning methods even when the spectrum is exponentially large.

5.3. Learning Parameterized Quantum Circuits

In this section, we study a concept class based on parameterized quantum circuits, which we call the PQC-based functions concept class. Then we describe an efficient quantum learner, based on the *Fourier coefficient extraction* algorithm proposed in the previous section. Finally, we show that this concept class is hard to learn classically, effectively proving a learning separation.

5.3.1. Concept Class Definition

We consider a family of parameterized circuits with Pauli encodings, as in Definition 5.1. We define x as the bitstring describing the fixed gates of the circuit. The parameterized gates have a known nature (P_l) but are parameterized by unknown parameters (α_l). This gives rise to the following concept class, which we call *PQC-based functions.*

Definition 5.2 (PQC-based functions concept class)
Consider a parameterized circuit U *on* n *qubits as defined in Definition 5.1,*

we note as x the bit string describing the fixed gates, and $\alpha \in [0,1]^d$ the input parameters of the circuit. We have:

$$U(x,\alpha)\left|0\right\rangle = \sum_{k\in\{0,1\}^n} \sum_{l\in\mathcal{L}'} a_{l,k}(x) e^{i\pi l\cdot\alpha} \left|k\right\rangle . \tag{5.10}$$

For a given observable O, we define the concept class $\mathcal{C}^{PQC}_{n,d}$, for a qubit number n and a number of unknown parameters d, which may scale with n.

$$\mathcal{C}^{PQC}_{n,d} \coloneqq \{c_\alpha : x \in \{0,1\}^n \to \left\langle 0\right| U(x,\alpha)^\dagger O U(x,\alpha) \left|0\right\rangle\}_{\alpha\in[0,1]^d} . \tag{5.11}$$

This concept class has been studied, in particular, its covering number has been formally upper-bounded [157], yielding a good generalization performance. As a consequence, as long as $d \in \mathrm{poly}(n)$, this class is PAC-learnable. Here we will be focusing on the possibility and impossibility of *efficient* PAC learning.

5.3.2. Efficient Quantum Learner

We consider the concept class as defined in Definition 5.2. In this subsection, we aim to prove that $\mathcal{C}^{PQC}_{n,\log n}$ is quantum efficiently PAC learnable as in Definition 2.2. To do so, we present an efficient quantum learning algorithm that uses the Fourier sampling procedure described in Section 5.2.

We are given a T sized dataset $\{x_t, y_t\}_{t\in\{1,\cdots,T\}}$ for an unknown fixed concept c_α for data sampled over $\mathcal{D}$, a maximum probability of failure as δ and the required accuracy as ε. The problem size is the number of qubits n, and we fix the number of parameters as $d = \log(n)$, each uploaded a constant number of times L. This setup yields a number of frequencies $m = |\mathcal{L}| = (4L+1)^d \in \Theta(\mathrm{poly}(n))$. We describe the training and inference stage of the proposed algorithm.

Training: First we choose the measurement accuracy ε_b for the Fourier coefficients. For each Fourier coefficient, a simple sample average strategy is used, requiring ε_b^{-2} shots (a quadratic improvement may be possible here). For each bitstring x_t specifying a circuit and for each frequency l, the coefficient $b_l(x_t)$ is retrieved using Corollary 5.2 up to additive error ε_b. This procedure requires the execution of mT/ε_b^2 poly-depth quantum circuits. The retrieved Fourier coefficients are stacked in the matrix $\hat{B}$ where the rows correspond to the frequencies indexed by l and the columns to the samples indexed by t. Similarly, we define the exact Fourier coefficients as B and the shot noise $E = B - \hat{B}$.

We have that $y = b(x) \cdot w(\alpha)$, where w is an unknown weight vector with $w_l(\alpha) := e^{i\pi l \cdot \alpha}$. In other words, the concept is linear with respect to B, and stacking all labels in a vector Y, we have $Bw = Y$. Our goal is to minimize the mean square error on unseen data. We look for a hypothesis function h that minimizes the error defined as.

$$\mathcal{L}_c(h) = \mathbb{E}_{x\sim D}[|h(x) - c_\alpha(x)|^2]\,. \tag{5.12}$$

We use a regression technique (LASSO, see details in Section 5.7.1) that yields a mean square error minimizer, with an additional constraint on the 1-norm of the weight vectors, which should be upper bounded by Λ_1. Minimizers returned by this regression technique belong to the following set of functions, which we define as our hypothesis class:

$$\mathcal{H} = \{b \xrightarrow{h_w} b \cdot w \mid \|w\|_1 \leq \Lambda_1\} \tag{5.13}$$

Using Theorem 5.6 with $\Lambda_1 = \|w\|_1 = m$ and $r_\infty = 1$ we know that in order to reach ε accuracy on the mean square error with $1 - \delta$ accuracy we can choose the parameters as follows, (noting that $\varepsilon_y = 0$).

$$T = \frac{16m^4\sqrt{2\log\left(\frac{2m}{\delta}\right)}}{\varepsilon^2},\ \varepsilon_b = \frac{0.2\varepsilon}{m} \tag{5.14}$$

We note that $T \in \tilde{\Theta}(\text{poly}(n, \varepsilon^{-1}, \delta^{-1}))$. In addition, the training stage of our algorithm outputs a weight vector $\hat{w}$ executing of $\tilde{\Theta}(m^7/\varepsilon^4) \subset \tilde{\Theta}(\text{poly}(n, \varepsilon^{-1}, \delta^{-1}))$ poly-depth quantum circuits. Therefore, both the number of samples and the runtime required to output a model are polynomial in n, ε^{-1}, and δ^{-1}.

Inference: Given a new datapoint $x_{t'}$, one retrieves the Fourier coefficients $b_{t'}$ using the algorithm in Section 5.2 execution of $m/\varepsilon_b^2 \in \tilde{\Theta}(m^3/\varepsilon^2) \subset \tilde{\Theta}(\text{poly}(n, \varepsilon^{-1}, \delta^{-1}))$ poly-depth circuits. Then using the weight vector $\hat{w}$ derived in the training phase, the model returns $\hat{y}_{t'} = \hat{w} \cdot \hat{b}(x_{t'})$. Theorem 5.6 guarantees that $|\hat{y}_{t'} - y_t| \leq \varepsilon$ with probability $1 - \delta$. The runtime of the model is polynomial in n, ε^{-1} and δ^{-1}.

This concludes the proof of our first main result, which we summarize in the following theorem.

Theorem 5.2
The PQC-based functions concept class $\mathcal{C}_{\log}^{PQC}$ *as in Definition 5.2 is effi-*

ciently quantum PAC learnable as in Definition 2.2.

5.3.3. Hardness of the learning problem

In this subsection, we prove that the PQC-based functions concept class is not classically efficient PAC learnable unless $\mathsf{BQP} \subset \mathsf{P/poly}$, for more on this class see Section 5.7.2.

We choose a promise BQP language $\mathcal{L}$, solved by $U \in \mathcal{U}(2^n)$, meaning that the sign of $\mathrm{Tr}\big[OU\,|x\rangle\langle x|\,U^\dagger\big]$ decides $\mathcal{L}$, where O is some simple observable. We define the concept class as in Definition 5.2 as the circuit U with added parameterized gates P_l at fixed locations with unknown parameters α_l.

The concept class contains at least one concept that is BQP-hard, for $\alpha = 0$. Lemma 2 in [22] (originally found in [66]) states that the efficient learnability of a concept class implies efficient evaluation of all concepts. However, unless $\mathsf{BQP} \subset \mathsf{P/poly}$, no classical algorithm can execute c_0 as it is BQP-hard. Therefore the concept class is not classically learnable if Conjecture 5.1 is true. This yields the following theorem.

Theorem 5.3
If $\mathsf{BQP} \not\subset \mathsf{P/poly}$ *then the PQC-based functions concept class Definition 5.2 is not classically efficient PAC learnable as in Definition 2.2.*

Theorem 5.3 and Theorem 5.2 together prove a separation between quantum and classical learners for the PQC-based functions problem. However, this case is rather contrived and does not have any direct physical application. In the next section, we adapt this result to when a circuit compiles a time evolution, which yields a more practical learning problem.

5.4. Learning Time Evolution

5.4.1. Concept Class Definition

Our main result is a learning separation for the Hamiltonian dynamics learning problem.

We consider a parameterized Hamiltonian $H(x, \alpha)$ the input is $x \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}^d$ are unknown parameters. We are interested in its time evolution for a fixed time τ, as $U_n(x, \alpha) = e^{i\tau(H(x,\alpha))}$.

This gives rise to the following concept class, which we call *Hamiltonian dynamics.*

Definition 5.3 (Hamiltonian dynamics concept class)
Consider a sequence of parameterized Hamiltonian $\{H_n(x, \alpha)\}_n$*, where each* H_n *operates on* n *qubits and is described by continuous parameters* $\alpha \in [0, 1]^d_n$ *and bitstrings* x *of length* s_n*. For a given sequence of observables* $\{O_n\}_n$*, we define the concept class* $\mathcal{C}^H_{n,d}$*, for* n *qubits,* d *parameters that may scale with* n*, and a fixed real number* τ *, resulting in a time evolution* $U_n(x, \alpha) = e^{i\tau H_n(x,\alpha)}$*, as follows,*

$$\mathcal{C}^H_{n,d} \coloneqq \{c_\alpha : x \in \{0,1\}^{s_n} \to \langle 0| U_n(x,\alpha)^\dagger O U_n(x,\alpha) |0\rangle\}_{\alpha \in [0,1]^d} . \tag{5.15}$$

5.4.2. Connection between the two concept classes

The Hamiltonian dynamics concept class can be related to the PQC-based functions concept class via Hamiltonian simulation. Given the class $\mathcal{C}^H_{n,\log n}$, by using Hamiltonian simulation on the underlying Hamiltonians (taking into account the parametrizations), we obtain parameterized circuits. The precision we use in Hamiltonian simulation dictates how closely functions in one class approximate the functions in the other in a precisely quantified way, and at the same time, the depth of the parameterized circuit.

We illustrate this with an example. Consider the Ising Hamiltonian on an arbitrary graph with a transverse field of unknown strength α. The arbitrary graph is described by bitstrings indicating whether an edge exists or not, $x_{i,j} = 1$ if $(i, j) \in E$.

$$H(x, \alpha) = \sum_{i,j} x_{i,j} Z_i Z_j + \alpha \sum_i X_i \tag{5.16}$$

The first order Trotterization of r steps, yields the following parameterized quantum circuit for the parameter $\alpha\tau/r$.

$$U_r(x, \alpha) = \left(\prod_{i,j} Z_i Z_j (x_{i,j}\tau/r) \prod_i X_i(\alpha\tau/r) \right)^r \tag{5.17}$$

This is illustrated in Figure 5.2.

We use this connection to show that the learning algorithm that learns $\mathcal{C}^{PQC}_{n,\log n}$ can also learn the time-dynamics class $\mathcal{C}^H_{n,\log n}$. In the rest of this section, we will adapt the results of the previous section to a quantum circuit approximating this evolution. As the quantum learner we devised previously is only efficient for polynomially sized spectrum, we investigate

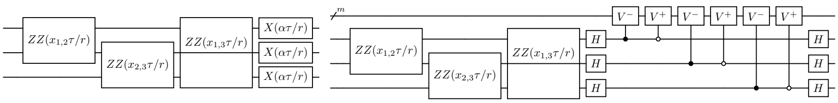


Figure 5.2.: Example of the Trotterization of the τ-time evolution of $H(x, \alpha) = \sum_{i,j} x_{i,j} Z_i Z_j + \alpha \sum_i X_i$ for three qubits in r step. The circuit presented is just one Trotter step and shall be repeated r times. (left) Compilation as a Pauli encoded parameterized quantum circuit as in Definition 5.1. (right) Fourier coefficient extraction applied to the original circuit.

circuit compilation such that this is guaranteed.

With the same strategy as that of the previous section, we show that the learning algorithm for the PQC-based functions concept class also works as an efficient learning algorithm for the Hamiltonian dynamics concept class. For completeness, we have derived the full proof in Section 5.7.6 for first order Trotterization. The key idea is that Hamiltonian simulation allows us to construct a good hypothesis class for our learning algorithm based on the concept class. In fact, as we explain later, all Hamiltonian simulation methods lead to function families with essentially the same learning characteristics.

Theorem 5.4
The Hamiltonian dynamics concept class $\mathcal{C}^H_{\log}$ as in Definition 5.3 is efficiently quantum PAC learnable as in Definition 2.2.

5.4.3. Hardness of the learning problem

In this subsection, we prove that the Hamiltonian dynamics concept class is not classically efficient PAC learnable unless $\mathsf{BQP} \subset \mathsf{P/poly}$.

Specifically, we will be proving that the special case of the Hamiltonian dynamics concept class, where the input is a bit string that specifies the initial state, and the Hamiltonian is parameterized only by α is hard.

For this, we use Lemma 3 in [22], based on [158], which is restated here.

Lemma 5.3
[from [22]] For any k-gate quantum circuit $U = U_k \cdots U_2 U_1$ acting on n qubits there exists a local Hamiltonian H such that for any n qubit initial state $|\psi\rangle$, we have

$$e^{iH\pi} |\psi\rangle |0\rangle = U |\psi\rangle |k\rangle \qquad (5.18)$$

We choose a promise BQP language $\mathcal{L}$, solved by $U \in \mathcal{U}(2^n)$, meaning that the sign of $\mathrm{Tr}\big[OU\,|x\rangle\!\langle x|\,U^\dagger\big]$ decides $\mathcal{L}$, where O is some simple observable. We define the $V(x)$ that prepares $|x\rangle$ from $|0\rangle$. We apply Lemma 5.3 and get $H(x)$ such that

$$e^{iH(x)\pi}\,|0\rangle\,|0\rangle = UV(x)\,|0\rangle\,|k\rangle = U\,|x\rangle\,|k\rangle \tag{5.19}$$

Choosing the observable $O' = O \otimes I$, we define the concept class as in Definition 5.3 for the hamiltonian $H(x,\alpha)$ with added parameterized Pauli terms P_l with unknown parameters α_l.

$$H(x,\alpha) = H(x) + \sum \alpha_l P_l \tag{5.20}$$

The concept class contains at least one concept that is BQP-hard, for $\alpha = 0$. Therefore, by the same argument as for the PQC-based functions concept class, using Lemma 2 in [22], this yields the following theorem.

Theorem 5.5
If $\mathsf{BQP} \not\subset \mathsf{P/poly}$ *then the Hamiltonian dynamics concept class Definition 5.3 is not classically efficiently PAC learnable as in Definition 2.2.*

5.5. Beyond log-many parameters

5.5.1. Exponentially large spectrum

So far we have considered concept classes where the number d_n of α parameters scales logarithmically with n, making them restricted. When instead of d_n scales polynomially, the cardinality of the spectrum $|\mathcal{L}|$ scales exponentially. The proposed algorithm is no longer efficient in general, as it would require performing regression in an exponentially large space. In addition, for complexity-theoretic reasons, devising a scheme that provably efficiently PAC-learns any setting with an exponentially large spectrum is not possible.

Specifically, in [159] it was proven that the learning of shallow classical circuits, even in the quantum PAC model (strictly stronger model than ours as the data in terms of a purification rather than samples from a distribution) is impossible under common complexity-theoretic assumptions, e.g. that ring-learning with errors cannot be done in polynomial time on a quantum computer. Polynomially-sized PQCs with polynomially many parameters can encode shallow classical circuits, and thus their efficient learnability would also imply the unlikely efficient quantum algorithms for learning with errors.

However, giving up on provable bounds, we can still devise a heuristic algorithm making use of the proposed feature map.

5.5.2. Kernel approach

We propose a kernel method approach as a heuristic. Consider the PQC-based functions concept class, for which we have $c_\alpha(x) = \sum_{l \in \mathcal{L}} b_l e^{i\pi\alpha \cdot l}$, or equivalently, for any $l \in \mathcal{L}$:

$$b_l(x) = \int_{(0,1)^d} c_\alpha(x) e^{-i\pi\alpha \cdot l} d\alpha \tag{5.21}$$

We define the kernel as,

$$k(x, x') = b(x) \cdot b(x')\,, \tag{5.22}$$

and make use of the quantum algorithm proposed in Section 5.2 to estimate kernel values. We provide more details on the circuit we use to do so efficiently in Section 5.7.7. Based on this, we build the $T \times T$ Gram matrix with $O(T^2)$ evaluations on a quantum computer. Finally, we can perform traditional kernel methods on a classical computer, for example, kernel ridge regression.

We discuss the limitations of this kernel approach and describe conditions under which they can be mitigated. In general, the dimension of the feature map of the resulting kernel is exponentially large. Therefore, the generalization performance is not guaranteed unless we have an exponential number of training samples. In fact, quantum kernels famously suffer from problems of exponential concentration [160]. A number of known factors yield exponential concentration, such as the expressivity of the data embedding, or global measurements. However, looking at the circuit producing the kernel evaluation in Figure 5.8, we argue that the kernel we propose is not immediately concerned by any of the known causes of exponential concentration in quantum kernels.

In fact, we prove that if the spectrum is sparse, with polynomially large support, the kernel ridge regression yields an efficient PAC learning algorithm in Section 5.7.7. It is possible to construct an artificial case where we get a frequency support that is only polynomially large for an a priori exponentially large spectrum. Consider the concept class Definition 5.2 with $d \in O(\mathrm{poly}(n))$, but such that most α parameters cancel each other out by construction, with for example $R_z(\alpha_s) Y R_z(\alpha_s)$. Suppose that only logarithmically many α survive these cancellations. While the spectrum is a priori exponentially large, it is in reality only polynomially large. The proposed kernel approach could efficiently learn this concept class, while no classical learner could unless $\mathsf{BQP} \subset \mathsf{P/poly}$.

5.5.3. Properties of the feature map

By Mercer's theorem, any kernel has a feature map associated with it. In the kernel approach that we propose, the feature map is simply $x \to b(x)$. That is, the feature map is directly the complex vector that represents the Fourier coefficients with respect to α of the function described by a bitstring x.

In this subsection, we argue that for the Hamiltonian dynamics concept class, the feature map associated with the kernel is in some sense equivalent for any circuit approximating the function. In particular, this means that as long as a given precision level is reached, the feature map is invariant with respect to the Hamiltonian time evolution technique. For simplicity, we propose to use Trotterization, where the depth scales polynomially with the error, but, for more optimal schemes [161, 162], the depth scales logarithmically with the error. We explain this in more detail below.

Given a bounded function $f : [0,1]^d \to \mathbb{C}$, it is always possible to define its Fourier coefficients as $b_l = \int_{(0,1)^d} f(\alpha) e^{i2\pi l \cdot \alpha} d\alpha$ for any $l \in \mathbb{Z}^d$. That is, any such function can be represented as a complex vector in an infinite-dimensional space. We have also seen that when a parameterized circuit is Pauli-encoded, it has a finite Fourier representation, that is, b is a finite-dimensional vector. We argue that if a Pauli-encoded parameterized quantum circuit approximates a function, its finite-dimensional vector approximates the infinite-dimensional one in the 2-norm. The Hausdorff-Young inequality [163] states that, for any two integer $p \in [1,2]$ and p' such that $\frac{1}{p} + \frac{1}{p'} = 1$, we have,

$$\Big(\sum_{l \in \mathbb{Z}^d} |b_l|^{p'} \Big)^{1/p'} \leq \Big(\int_{(0,1)^l} |f(\alpha)|^p \, d\alpha \Big)^{1/p}. \tag{5.23}$$

Consider an exact time evolution yielding a concept c_α (see Definition 5.3). It is approximated up to $\varepsilon/2$ by a first circuit yielding the functions c' and another circuit yielding the function c''. We note their Fourier decomposition b' and b'' respectively. Then, using the Hausdorff-Young inequality with $p = p' = 2$ we get an upper bound of the 2-norm of the Fourier decomposition of the difference $c' - c''$ as its infinite norm, itself bounded by ε, as follows,

$$\|b'_l - b''_l\|_2 \leq \Big(\int_{(0,1)^d} |(c' - c'')(\alpha)|^2 \, d\alpha \Big)^{1/2} \leq \varepsilon \, . \tag{5.24}$$

This concludes that feature maps approximating the same quantum function are ε-close in the 2-norm metric space. This analysis implies that the learning and generalization properties of the kernel will not in any significant way depend on which Hamiltonian simulation technique is used, i.e. which hypothesis function is chosen for the quantum learning algorithm.

5.6. Discussion

5.6.1. Cardinality of the Concept class

In the first work demonstrating quantum-classical learning separations for quantum functions the concept classes were polynomially sized, which meant a brute-force algorithm checking all hypotheses is efficient [21]. In [22, 164] first examples of learning with an exponentially-sized concept class were introduced.

The concept classes we study here, the PQC-based functions and the Hamiltonian dynamics concept classes, are indexed by continuous parameters, and each parameter setting mathematically specifies a different function (up to periodicity concerns), and thus the class is a continuum. However, we note that we are interested in approximations, that is, finding functions which agree with the true function on a $1 - \delta$ fraction of the inputs when sampled from the input distribution. The question then arises of what the effective size of this function family is. Precisely, we are interested in the hypothesis family $\mathcal{H}$, such that for each $c \in \mathcal{C}$ there exists $h_c \in H$ such that h_c is ε-close to c in the PAC sense as in Definition 2.2. H can depend on ε and the input distribution $\mathcal{D}$.

If $\mathcal{H}$ is polynomially sized in n and its elements can be efficiently constructed and evaluated, then this would allow for a brute-force type algorithm for our learning task in the log-parameter case. We note that demanding that $\mathcal{H}$ is a subset of $\mathcal{C}$ we obtain the notion of ε-packing of the concept class, which is closely related to covering numbers, and both are quite well understood as quantities governing, for example, generalization bounds [157].

For our concept class PQC-based functions, to the best of our effort to obtain a tight bound (see Section 5.7.4) we can find a super-polynomial grid of functions which attains epsilon-packing, upper bounding the smallest effective hypothesis size to $O(n^{\log\log(n)})$. We conjecture that this log log scaling in the exponent may be an artefact of the bounding method and that, in fact, the concept classes with d in $O(\log(n))$ allow for a polynomial-sized effective class and a more direct learning algorithm. However the key question of whether this is true and whether these hypotheses can be efficiently found and evaluated remains open.

Therefore, the method we provide is not brute force, but the arguments above raise the question of whether a brute force method could also be possible for our learning task. Either way, we emphasize that a learning separation persists.

5.6.2. Conclusions

In this chapter, we have proposed an algorithm that efficiently prepares a state representing the Fourier decomposition of some quantum functions. We have defined a concept class based on parameterized quantum circuits that can be efficiently PAC learned using a quantum computer and standard regression techniques. As this first concept class is not physically relevant, we built on it and propose a concept class based on a Hamiltonian time evolution and show that it is also quantum-efficient PAC learnable. We do so by applying the previous result to the Trotterized time evolution, but later discuss that any quantum simulation technique would yield similar performance. We also show that both classes cannot be PAC-learned efficiently by any classical algorithm unless $\mathsf{BQP} \subset \mathsf{P/poly}$, effectively proving a learning separation. Both concept classes have polynomially sized feature space, which yields a weaker learning separation in comparison with brute force approaches [21]. We finally discuss regimes in which a priori exponentially large feature space could remain efficiently PAC-learnable and avoid exponential concentration issues [160].

5.7. Appendix

5.7.1. LASSO regression

LASSO (Least Absolute Shrinkage and Selection Operator) regression is a linear regression method [165] with an added regularization on the 1-norm of the weight vector to the loss function. This enforces sparsity in the estimated coefficients. Given an input matrix $X \in \mathbb{R}^{m\times T}$ and label vector $y \in \mathbb{R}^m$, the LASSO estimator solves a mean square error minimization with a constraint of the norm 1 of the weight vector.

$$\hat{w} = \arg\min_w \|y - Xw\|_2^2 \text{ subject to } \|w\|_1 \leq \Lambda_1,$$

where $\Lambda_1 > 0$ controls the trade-off between sparsity and model fit. This property makes LASSO particularly useful in high-dimensional settings where many predictors may be irrelevant. In Section 5.7.1 we prove the following theorem about the robustness of the LASSO regressor in the presence of perturbations.

Theorem 5.6
Consider a dataset of size T, denoted $\{(\hat{b}_t, \hat{y}_t)\}_{1 \leq t \leq T}$, generated from a linear model with an unknown weight vector $w \in \mathbb{R}^m$, and affected by perturbations, as follows:

$$b_t \cdot w = y_t, \tag{5.25}$$

$$\hat{b}_t = b_t + \eta_{b,t}, \tag{5.26}$$

$$\hat{y}_t = y_t + \eta_{y,t}, \tag{5.27}$$

where $\|\eta_{b,t}\|_\infty$ and $\|\eta_{y,t}\|_\infty$ are bounded by ε_b and ε_y, respectively. Here, m denotes the dimension of b and we know r_∞ such that $\|b\|_\infty \leq r_\infty$. Running the LASSO algorithm with the constraint that $\|w\|_1 \leq \Lambda_1$ on this dataset, with probability at least $1 - \delta$, yields a model h^* such that the true risk (training error + generalisation error) is at most ε, provided that

$$T \geq \frac{(2\Lambda_1 r_\infty)^4 \sqrt{2\log\left(\frac{2m}{\delta}\right)}}{\varepsilon^2}. \tag{5.28}$$

$$\Lambda_1 \varepsilon_b \leq 0.2\varepsilon \tag{5.29}$$

$$\varepsilon_y \leq 0.5\varepsilon \tag{5.30}$$

We prove Theorem 5.6 below. First, we state the well-known generalization bound for the LASSO algorithm. Then we prove a lemma that bounds the empirical risk, and then we combine these two to bound the true risk.

Theorem 5.7
Let $\mathcal{X} \subseteq \mathbb{R}^m$ and

$$\mathcal{H} = \{b \in \mathcal{X} \mapsto w \cdot b : \|w\|_1 \leq \Lambda_1\}.$$

Let $S = ((b_1, y_1), \ldots, (b_T, y_T)) \in (\mathcal{X} \times \mathcal{Y})^T$. Let $\mathcal{D}$ denote a distribution over $\mathcal{X} \times \mathcal{Y}$ according to which the training data S is drawn. Assume that there exists $r_\infty > 0$ such that for all $b \in \mathcal{X}$, $\|b\|_\infty \leq r_\infty$, and $M > 0$ such that $|h(b) - y| \leq M$ for all $(b, y) \in \mathcal{X} \times \mathcal{Y}$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, each of the following inequalities holds for all $h \in \mathcal{H}$:

$$\mathcal{R}(h) \leq \hat{\mathcal{R}}_S(h) + 2r_\infty \Lambda_1 M \sqrt{\frac{2\log(2m)}{T}} + M^2 \sqrt{\frac{\log(\delta^{-1})}{2T}},$$

where $\mathcal{R}(h) = \mathbb{E}_{(b,y)\sim\mathcal{D}} \left[|h(b) - y|^2\right]$ is the prediction error for the hypothesis h, and $\hat{\mathcal{R}}_S(h)$ is the training error of h on the training data S.

Lemma 5.4 (Upper Bound on Empirical Risk under Bounded Perturbations)
Let $h^*(x) = \mathbf{w}^* \cdot \hat{\mathbf{b}}(x)$ denote the hypothesis returned by the LASSO algorithm, trained on a dataset with perturbed features $\hat{\mathbf{b}}(x)$ and noisy labels $\hat{y} = y + \eta_y$. Suppose the true labeling function is linear, i.e., $y = \mathbf{w} \cdot \mathbf{b}(x)$, for some weight vector $\mathbf{w} \in \mathbb{R}^m$, and assume that $\|\mathbf{w}\|_1 \leq \Lambda_1$. Furthermore, assume that the unperturbed features satisfy $\|\mathbf{b}(x)\|_\infty \leq r_\infty$, the perturbation on the features is bounded by $\|\hat{\mathbf{b}}(x) - \mathbf{b}(x)\|_\infty \leq \varepsilon_b$, and the additive noise on the labels is bounded as $|\eta_y| \leq \varepsilon_y$. If the LASSO optimization problem is solved approximately such that the empirical risk of h^* is within $\varepsilon_3/2$ of the optimal empirical risk over all weight vectors with ℓ_1-norm at most Λ_1, then the empirical risk of the resulting hypothesis satisfies

$$\hat{\mathcal{R}}_S(h^*) = \frac{1}{T} \sum_{t=1}^{N} \left(h^*(x_t) - y_t\right)^2 \leq \left(\Lambda_1 \varepsilon_b + \varepsilon_y\right)^2 + \frac{\varepsilon_3}{2}.$$

Proof. For any function g, the training error is defined as

$$\hat{\mathcal{R}}(g) = \frac{1}{T} \sum_{t=1}^{T} |g(x_t) - y_t|^2.$$

Now let $\mathbf{w}^*$ be the optimal vector that the LASSO algorithm outputs, i.e.,

$$\mathbf{w}^* = \arg\min_{\|\mathbf{w}\|_1 \leq \Lambda_1} \left(\frac{1}{T} \sum_{t=1}^{T} \left| \mathbf{w} \cdot \hat{\mathbf{b}}(x_t) - y_t \right|^2 \right).$$

It is clear that for any other $\mathbf{w}' \neq \mathbf{w}^*$ with bounded norm (i.e. $\|\mathbf{w}'\|_1 \leq \Lambda_1$),

$$\frac{1}{N}\sum_{i=1}^{N}\left|\mathbf{w}^* \cdot \hat{\mathbf{b}}(x_i) - y_i\right|^2 \leq \frac{1}{N}\sum_{i=1}^{N}\left|\mathbf{w}' \cdot \hat{\mathbf{b}}(x_i) - y_i\right|^2.$$

Let $\mathbf{w}$ be the true weight vector in the true labeling function $h(x) = \mathbf{w}\cdot\mathbf{b}(x)$. Using this, we can bound the training error for the function $h'(x) = \mathbf{w}'\cdot\hat{\mathbf{b}}(x)$. Let

$$t^* = \arg\max_{0\leq t\leq T}|\mathbf{w} \cdot \hat{\mathbf{b}}(x_t) - y_t|^2$$

be the index of the training data point that maximizes the loss. Then:

$$\begin{aligned}
\hat{\mathcal{R}}(h') &= \frac{1}{T}\sum_{t=1}^{T}|\mathbf{w}' \cdot \hat{\mathbf{b}}(x_t) - y_t|^2 \\
&\leq \frac{1}{T}\sum_{t=1}^{T}|\mathbf{w} \cdot \hat{\mathbf{b}}(x_t) - y_t|^2 \\
&\leq |\mathbf{w} \cdot \hat{\mathbf{b}}(x_{t^*}) - y_{t^*}|^2 \\
&\leq \Big(|\mathbf{w} \cdot \hat{\mathbf{b}}(x_{t^*}) - \mathbf{w} \cdot \mathbf{b}(x_{t^*})| + |\mathbf{w} \cdot \mathbf{b}(x_{t^*}) - y_{t^*}|\Big)^2 \\
&\leq (\Lambda_1\varepsilon_b + \varepsilon_y)^2.
\end{aligned}$$

Now consider the function $\hat{h}(x) = \hat{\mathbf{w}} \cdot \hat{\mathbf{b}}(x)$, where $\hat{\mathbf{w}}$ is obtained by minimizing the training error such that its empirical risk is at most $\varepsilon_3/2$ worse than the minimum. That is, we allow for a suboptimal solution. Then the empirical risk of $\hat{h}$ satisfies:

$$\hat{\mathcal{R}}(\hat{h}) \leq (\Lambda_1\varepsilon_b + \varepsilon_y)^2 + \frac{\varepsilon_3}{2}.$$

□

The proof of Theorem 5.7 follows from the prediction error of the LASSO algorithm and the bound on empirical risk derived in Lemma 5.4. First, we address M, which is the upper bound on the absolute prediction error. In the noisy case where LASSO receives $\hat{b}$ as input and the labels are also noisy, we have:

$$|\hat{h}(\hat{b}) - \hat{y}| \leq M.$$

Now:

$$\begin{aligned}|\hat{h}(\hat{b}) - \hat{y}| &= |\hat{b}\cdot\hat{w} - \hat{y}| \\ &= |\hat{b}\cdot\hat{w} - y - \eta_y| \\ &\leq |\hat{b}\cdot\hat{w}| + |y + \eta_y| \\ &\leq \|\hat{b}\|_\infty \|w\|_1 + |y| + |\eta_y|,\end{aligned}$$

where the last inequality follows from Hölder's inequality. The error on the labels is bounded, so $|\eta_y| \leq \varepsilon_y$, and $\|w\|_1 \leq \Lambda_1$. Since the true label is $y = b \cdot w$, and under the assumptions $\|b\|_\infty \leq r_\infty$ and $\|w\|_1 \leq \Lambda_1$, we get $|y| \leq \|b\|_\infty \|w\|_1 \leq r_\infty \Lambda_1$. Similarly, since $\|\hat{b}\|_\infty \leq r_\infty + \varepsilon_b$, and assuming $\|\hat{w}\|_1 \leq \Lambda_1$, we obtain the bound:

$$M := \Lambda_1(2r_\infty + \varepsilon_b) + \varepsilon_y.$$

Using this lemma, we can rewrite the generalization bound as:

$$\mathcal{R}(\hat{h}) \leq (\Lambda_1\varepsilon_b + \varepsilon_y)^2 + \frac{\varepsilon_3}{2} + 2r_\infty\Lambda_1 M\sqrt{\frac{2\log(2m)}{T}} + M^2\sqrt{\frac{\log(\delta^{-1})}{2T}},$$

where $r_\infty = r_\infty + \varepsilon_b$. To bound the prediction error above by

$$\varepsilon = (\Lambda_1\varepsilon_b + \varepsilon_y)^2 + \varepsilon_3,$$

it suffices to choose T such that:

$$2r_\infty\Lambda_1 M\sqrt{\frac{2\log(2m)}{T}} + M^2\sqrt{\frac{\log(\delta^{-1})}{2T}} \leq \frac{\varepsilon_3}{2},$$

and substituting for M and r_∞, solving for T, gives the sample complexity:

$$T \geq \frac{(\Lambda_1(2r_\infty + \varepsilon_b) + \varepsilon_y)^4\sqrt{2\log(2m/\delta)}}{\varepsilon_3^2}.$$

By setting $\Lambda_1\varepsilon_b = 0.2\varepsilon$, $\varepsilon_y = 0.5\varepsilon$, and $\varepsilon_3 = 0.4\varepsilon$, the prediction error is bounded by

$$(\Lambda_1\varepsilon_b + \varepsilon_y)^2 + \varepsilon_3 \leq \varepsilon,$$

provided that

$$T > \frac{(2\Lambda_1 r_\infty)^4\sqrt{2\log(2m/\delta)}}{\varepsilon^2}.$$

which concludes the proof of Theorem 5.6.

5.7.2. Complexity assumption

In this chapter, we present learning separation statements that rely on the following widely believed conjecture.

Conjecture 5.1
$\mathsf{BQP} \not\subset \mathsf{P/poly}$.

This is a weaker conjecture than the following one used in previous works.

Conjecture 5.2
There exists a distribution $\mathcal{D}$ *such that* $\mathsf{BQP} \not\subset \mathsf{HeurP}^{\mathcal{D}}/\mathsf{poly}$.

Nonetheless, this too is believed to be true: the discrete logarithm problem or factoring are not believed to be in $\mathsf{HeurP/poly}$. For these problems, the average-case to worst-case reductions imply that if either problem is efficiently solvable heuristically, then it is also efficiently solvable in the worst case as well. This does not hold for BQP functions in general, so Conjecture 1 is indeed weaker. Still, in general, by Lemma 3 in [21] if there exists a single $\mathcal{L} \in \mathsf{BQP}$ that is not in $\mathsf{HeurP/poly}$ under some distribution, then for every BQP-complete problem, there exists a distribution under which the problem is not in $\mathsf{HeurP/poly}$.

Lemma 5.5 (from [21])
If there exists a $(L, \mathcal{D}) \notin \mathsf{HeurP/poly}$ *with* $L \in \mathsf{BQP}$, *then for every* $L' \in \mathsf{BQP\text{-}complete}$ *there exists a family of distributions* $\mathcal{D}' = \{\mathcal{D}'_n\}_{n\in\mathbb{N}}$ *such that* $(L', \mathcal{D}') \notin \mathsf{HeurP/poly}$.

5.7.3. Fourier coefficient extraction algorithm

Fourier representation of parameterized circuits

In this section, we prove the Theorem 5.1 and describe the algorithm $\mathcal{A}$. As mentioned in the main text, without loss of generality, we consider that every circuit as defined in Definition 5.1 uses strings of identity and Z matrices encoding. Indeed, if encoding with X or Y appear they can be changed to Z with local unitary gates, which can be absorbed in the set of fixed gates.

Description of the algorithm

The input to the algorithm $\mathcal{A}$ is a description of the n-qubit circuit as an alternating sequence between sets of fixed gates resulting in unitaries $\{U_s\}_{0\leq s\leq dL}$ and encoding gates

$$\{U_{s,j}(\alpha_j) \coloneqq \exp(i\pi\alpha_j \prod_{1\leq k\leq n} Z_k^{b_{s,j,k}})\}_{s,j\in[1,L]\times[1,d]}$$

where j describe the index of the dimension of the data that is encoded, and $b_{s,j}$ is the bitstring of the qubits affected by the encoding gate.

The output of the algorithm $\mathcal{A}$ is a description of a circuit on a larger number of qubits. Registers are added to the existing circuit register to keep track of frequencies of α on each of the d dimensions, each uploaded L times. The number of qubits needed are:

$$n = d\lceil\log(1+2L)\rceil \tag{5.31}$$

For simplicity we assumed that all variables are uploaded the same number of times but, it is possible that it is not the case, in which variables the number of qubits required are $n = \sum_j\lceil\log(1+2L_j)\rceil$.

There is also a single additional ancillary qubit that is used to compute parities. This ancillary qubit is not necessary, not having it occurs an overhead exponential in the locality of Pauli strings in the data encoding gates. Therefore, in total, there are n_T qubits as follows:

$$n_T = n + d\lceil\log(1+2L)\rceil \tag{5.32}$$

We name the registers $f = f_0\cdots f_j$ for the frequency registers, a for the ancillary qubit, and c for the circuit register. A gate G applied to the register r will be written as $G^{(r)}$. Control on the $|1\rangle$ ($|0\rangle$) state are written as C ($\bar{C}$). The frequency registers are also indexed by negative numbers. For the frequency register we define the increment (decrement) gate with unitary matrix V_+ ($V_-{=}V_+^\dagger$) ensuring unitarity with a circular condition as such:

$$V_+\left|k\right\rangle = \left|k+1\right\rangle, \forall -2L \leq k < 2L \tag{5.33}$$

$$V_+\left|2L\right\rangle = \left|-2L\right\rangle \tag{5.34}$$

The algorithm $\mathcal{A}$ returns a sequence of gates that follows that of the original circuit, where fixed gates are unchanged and applied to the c register, and encoding gates $\{j, b_{s,j}\}$ are transformed as follows. For each qubit affected non trivially by the encoding gate (that is, with a Z_k generator when $b_{s,j,k} = 1$) the parity is computed on the ancillary qubit, with a sequence of CNOT gates as follows:

$$D(s, j) = \prod_{k|b_{s,j,k}=1} C^{(c_k)} X^{(a)} \tag{5.35}$$

At this stage, the ancillary encodes the parity of the sub-bitstring for indexes $b_{s,j}$. Then the j-th frequency register, corresponding to the dimension of the input being encoded, is acted on based on the parity encoded in the ancillary as such:

$$G(j) = \left(C^{(a)} V_+^{(f_j)}\right) \left(\bar{C}^{(a)} V_-^{(f_j)}\right) \tag{5.36}$$

Effectively, the subspace where the parity of the substring is even sees an increment in the frequency of the input being encoded, while the other subspace sees a decrement. Finally, the ancillary qubit is reset to be reused later with $D(s, j)^\dagger = D(s, j)$.

The output of the algorithm is an alternating sequence between sets of gates being unchanged from fixed gates resulting in unitaries $\{U_{s,j}^{(c)}(\alpha_j)\}_{s,j}$ and gates replacing encoding gates as $\{D(s, j)G(j)D(s, j)\}_{s,j}$.

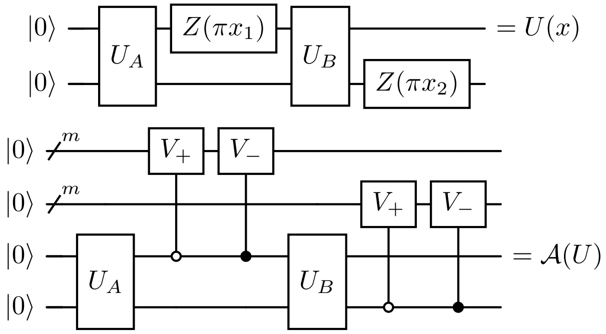


Figure 5.3.: Illustration of the *Fourier coefficient extraction* algorithm for multiple inputs

Proof of the algorithm

Finally, we prove that the algorithm $\mathcal{A}$ indeed outputs what it is promised to return. That is, given U such that

$$|\phi\rangle = U|0\rangle = \sum_{k\in\{0,1\}^n} \sum_{l\in\mathcal{L}'} a_{l,k} e^{i\alpha\cdot l} |k\rangle \ , \tag{5.37}$$

then it returns,

$$|\phi'\rangle = \mathcal{A}(U)|0\rangle = \sum_{k\in\{0,1\}^n} \sum_{l\in\mathcal{L}'} a_{l,k} |l\rangle^{(f)} |0\rangle^{(a)} |k\rangle^{(c)} \ . \tag{5.38}$$

For the rest of the proof, we shorten the indices k and l for easier notation. This is an induction proof in three steps:

1. Initial step : This is trivial, if $|\phi\rangle = |0\rangle$ then $|\phi'\rangle = \mathcal{A}(I)|0\rangle = |0\rangle^{(f)} \otimes |0\rangle^{(a)} \otimes |0\rangle^{(c)}$.
2. Fixed gate step: Suppose that the algorithm works as intended, before the application of a set of fixed gates with unitary V. Let us show that as algorithm $\mathcal{A}$ applies $V' = I^{(f)} \otimes I^{(a)} \otimes V^{(c)}$, it yields the correct state. It is easy to see that $V|\phi\rangle = \sum_l (\sum_k a_{l,k} V|k\rangle) e^{i\alpha\cdot l}$, and $V'|\phi\rangle = \sum_l (\sum_k a_{l,k} V|k\rangle^{(c)}) |0\rangle^{(a)} |l\rangle^{(f)}$.
3. Encoding gate step: Suppose that the algorithm works as intended before the application of an encoding gate $\{j, b_{s,j}\}$. Let us prove that applying $D(s,j)G(j)D(s,j)$ yields the correct state. We prove this below.

If these three steps are true, then by induction, the algorithm is correct.

Proof of step 3. The effect of the encoding gate $\{s, b_{s,j}\}$ on the state $|\phi\rangle$ is as follows

$$e^{i\alpha_j} \textstyle\prod_{k'} Z_{k'}^{b_{s,j,k'}} \displaystyle |\phi\rangle = \sum_k \sum_l a_{l,k} e^{i\alpha_j p(k,b_{s,j})} |k\rangle \, e^{i\alpha\cdot l} \tag{5.39}$$

$$= \sum_k \sum_l a_{l,k} |k\rangle \, e^{i\alpha\cdot(l + e_j p(k,b_{s,j}))} \tag{5.40}$$

Where $p(k, b_{s,j})$ is the parity of the substring of k with indices $b_{s,j}$ as follows $p(k, b_{s,j}) = (-1)^{\sum_{k'} k_{b_{s,j,k'}}}$. For example if $k = 010110$ and $b_{s,j} = 015$, the substring is 010, with odd parity and $p = -1$. We also write e_r as the vector such that only the r-th element is 1 and the rest is 0.

On the other hand, for the $\mathcal{A}(U)$ computation we have:

$$D(s,j)\,|\psi'\rangle \quad = \quad \sum_k \sum_l a_{l,k}\,|l\rangle^{(f)}\,(X^{\sum_{k'} k_{b_{s,j,k'}}}\,|0\rangle^{(a)})\,|k\rangle^{(c)} \tag{5.41}$$

$$G(j)D(s,j)\,|\psi'\rangle = \sum_k \sum_l a_{l,k}\,|l + e_j p(k, b_{s,j})\rangle^{(f)} \\ (X^{\sum_{k'} k_{b_{s,j,k'}}}\,|0\rangle^{(a)})\,|k\rangle^{(c)} \tag{5.42}$$

$$D(s,j)G(j)D(s,j)\,|\psi'\rangle \;=\; \sum_k \sum_l a_{l,k}\,|l + e_j p(k, b_{s,j})\rangle^{(f)}\,|0\rangle^{(a)}\,|k\rangle^{(c)} \tag{5.43}$$

The states correspond to each other, which concludes the proof to the iterative step 3.

Frequency representation of expectation values

So far, we have seen how to get the Fourier representation of a quantum reuploading circuit, but what about quantum function, that is, the expectation values of some observable? In this section, we prove Corollary 5.1 and Corollary 5.2. We explain how to do this for Pauli observables and extend this result to linear combinations of Pauli observables and projectors.

Pauli observable

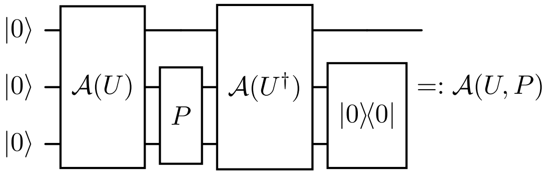


Figure 5.4.: Illustration of the quantum function evaluation algorithm

Without loss of generality (using local changes of basis), the Pauli string measurement can be considered a Z string and therefore $|\phi(\alpha)\rangle$ has the following form (where $p(P,k)$ is a parity function).

$$\begin{aligned} f(\alpha) &= \langle P\rangle(\alpha) && (5.44)\\ &= \sum_{-L\leq l',l\leq L}\sum_{k}(-1)^{p(P,k)}a_{l,k}a^{*}_{l',k}e^{i(l-l')\cdot\alpha} && (5.45)\\ &= \sum_{-2L\leq l\leq 2L} b_l e^{il\cdot\alpha} && (5.46)\end{aligned}$$

We define the state $|\phi(\alpha),P\rangle := U(\alpha)^{\dagger}PU(\alpha)|0\rangle$. It has the following amplitude for $|0\rangle$.

$$\langle 0|\phi(\alpha),P\rangle = \langle 0|U(\alpha)^{\dagger}PU(\alpha)|0\rangle = \sum_{-2K\leq k\leq 2K} b_k e^{ik\alpha} \tag{5.47}$$

Using the algorithm $\mathcal{A}$ on the state $|\phi(\alpha),P\rangle$ we can lose the dependence on α and efficiently get the Fourier representation state:

$$|\phi,P\rangle = \sum_{-2K\leq k\leq 2K} b_k|k\rangle|0\rangle + \text{trash}\,. \tag{5.48}$$

Post-selecting on $|0\rangle$ has a success probability $\sum|b_k|^2$. The full algorithm is illustrated in Figure 5.4.

Linear combination of Pauli observables

In this subsection, we extend the procedure above to a more generic observable, more specifically, a linear combination of polynomially many Pauli strings. This can be done using a Linear Combination of Unitaries approach. Supposing that the observable of interest may be decomposed as follows

$$O = \sum_h \beta_h P_h \tag{5.49}$$

Following the procedure described previously, one may prepare the states $|\phi,P_h\rangle$ for each P_h. By the linearity of the expectation value, adding them yields the expectation value of the full observable.

$$|\phi,O\rangle = \sum_h \beta_h|\phi,P_h\rangle \tag{5.50}$$

The addition may be done using a linear combination of unitary approach, which requires an additional register with a number of qubits logarithmic in the number of terms in the observable, and the preparation of the following state.

$$V_\beta |0\rangle = \frac{1}{\|\beta\|}\sum_h \beta_h |h\rangle \tag{5.51}$$

It also requires post-selection, which causes overhead in complexity (polynomial for reasonable observables). Once the state is prepared, the same sampling and post-processing may be implemented.

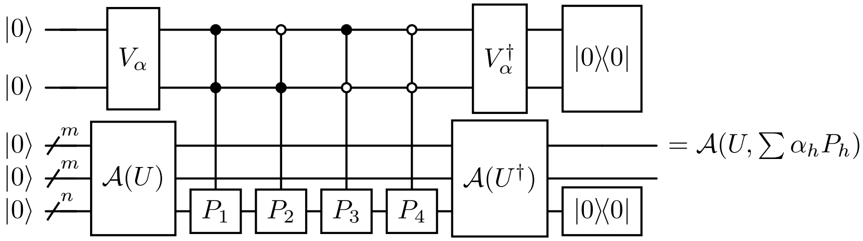

$|0\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
m
m
n
V_α
$\mathcal{A}(U)$
P_1
P_2
P_3
P_4
$V_\alpha^\dagger$
$\mathcal{A}(U^\dagger)$
$|0\rangle\langle 0|$
$|0\rangle\langle 0|$
$= \mathcal{A}(U, \sum \alpha_h P_h)$

Figure 5.5.: Illustration of the quantum function evaluation algorithm for arbitrary observables

Probabilities, or projectors observables

Consider a circuit U yielding a pure state $U|0\rangle = |\phi\rangle = \sum_k \phi_k |k\rangle$. Its density matrix is $\rho = \sum_{k,l} \phi_k \phi_l^* |k\rangle\langle l|$. Defining the conjugate circuit as U^*, we have

$$(U \otimes U^*)(|0\rangle \otimes |0\rangle) = \sum_{k,l} \phi_k \phi_l^* |k\rangle \otimes |l\rangle \tag{5.52}$$

Therefore, if one wanted to retrieve the probability of measuring $|0\rangle$ of circuit U one could retrieve the amplitude as above. This yields the procedure illustrated in Figure 5.6 to retrieve the coefficients for the observable $|0\rangle\langle 0|$, that is the probability of measuring $|0\rangle$ on the original circuit.

Extracting Fourier coefficients

Once the Fourier decomposition is available, one may be interested in retrieving the value coefficients. Suppose we have the following state as

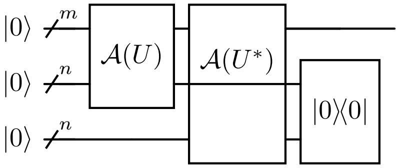


Figure 5.6.: Illustration of the quantum function evaluation algorithm for a projector

an output.

$$\mathcal{A}(U,O)\left|0\right\rangle\left|0\right\rangle = \left|\phi,O\right\rangle = \sum_{-2K\leq k\leq 2K} b_k \left|k\right\rangle\left|0\right\rangle + \text{trash} \tag{5.53}$$

Suppose one is interested in the coefficient of frequency l. We call V_l any unitary such that $\left|l\right\rangle = V_l\left|0\right\rangle$, we have

$$b_l = \left\langle 0\right|\left\langle 0\right|(V_l^\dagger \otimes I)\mathcal{A}(U)\left|0\right\rangle\left|0\right\rangle \tag{5.54}$$

We use a Hadamard test (which can be improved as in [166]) to extract the real and imaginary part of this coefficient as in Figure 5.7. In addition, because the function is real, we have $b_k = b_{-k}$. Therefore, in case there is a polynomial number of frequencies, one may retrieve all of them efficiently, and create a classical surrogate for the quantum function.

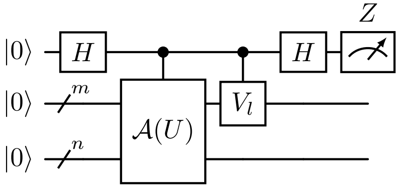


Figure 5.7.: Illustration of the *Fourier coefficient extraction* algorithm

5.7.4. Cardinality of the concept class

First we will find a relatively easy bound for the covering number of the following set of functions:

$$\mathcal{C} = \{c_\alpha : x \to \sum_{-L\leq l\leq L} b_l(x)e^{il\cdot\alpha}, \|c_\alpha\|_\infty \leq 1\}_{\alpha\in[0,1]^d} \tag{5.55}$$

First we construct a covering net using the smoothness of the function with respect to its index α. The partial derivative with respect to a single parameter is as follows,

$$|\partial_{\alpha_k} c_\alpha(x)| \leq |\sum_l il_k b_l(x)e^{il\cdot\alpha}| \leq L\sum_l |b_l(x)e^{il\cdot\alpha}| \leq L\|b(x)\|_1\,. \tag{5.56}$$

We can use it to bound the 1-norm of the gradient as,

$$\|\nabla_\alpha c_\alpha(x)\|_1 \leq dL\|b(x)\|_1\,. \tag{5.57}$$

We consider a regular grid with M divisions for each of the d dimensions. We call this grid $\mathcal{A} = \{\alpha_g\}_g$, with $|\mathcal{A}| = M^d$. Therefore, each α will be at least $\|\delta_\alpha\| := 1/M$ close to a grid point. We have

$$\begin{aligned}\|c_\alpha - c_{\alpha_g}\|_\infty &\leq \nabla_\alpha c_\alpha \cdot (\alpha - \alpha_g) & (5.58)\\ &\leq \|\nabla_\alpha c_{\alpha_g}\|_1 \|\delta_\alpha(\alpha)\|_\infty & (5.59)\\ &\leq dL\|b(x)\|_1/M\,. & (5.60)\end{aligned}$$

To guarantee the error is lower than ε for all α it is sufficient to choose

$$M = dL\|b(x)\|_1/\varepsilon\,. \tag{5.61}$$

We are interested in the scaling of the grid when $d \in O(\log(n))$ and $L \in O(1)$. For all x we have $\|b(x)\|_2 \leq 1$ and therefore $\|b(x)\|_1 \leq (2L+1)^{d/2} \in O(\mathrm{poly}(n))$. This yields a number of grid points scaling as

$$G = (dL\|b(x)\|_1/\varepsilon)^d \in O(n^{\log(n)})\,. \tag{5.62}$$

If we leave this function family $\mathcal{C}$, we can apply the approach of [157] states bounds on covering numbers of shot-based expectation values of parameterized quantum circuits. This allow us to get a smaller hypothesis class of proxy functions approximating the functions of the concept class. Theorem 3 in the supplementary material of [157] upper bounds the covering number for a PQC where d parameters (T in the original paper) are reuploaded L times (M in the original paper). In order to get the expectation values up to η additive accuracy, the number of reuploadings is multiplied by η^{-2} Using $d \in O(\log n)$ and $L \in O(1)$ we get the following scaling:

$$G \in O\left(\left(\frac{dL}{\varepsilon\eta^2}\right)^d\right) \subseteq O(n^{\log\log n}) \,. \tag{5.63}$$

Although this scaling is closer to a polynomial bound, it is not polynomial, but this might be an artefact of the proof. The key question of whether there exist a polynomial and constructable set of of hypothesis functions $\mathcal{H}$ that ε-covers $\mathcal{C}$ remains open.

5.7.5. Alternative Oracle-based algorithms

Considering the settings of Section 5.2, with a $U(\alpha)$ defined as above, suppose that instead of the specification of the parameterized quantum circuit U as a sequence of gates, we are given an oracle O_U such that for a m-binary decomposition over the d-dimensional parameter α, we have

$$|\alpha\rangle \otimes |\psi\rangle \xrightarrow{O_U} |\alpha\rangle \otimes U(\alpha)\,|\psi\rangle \ .$$

Applying this oracle to the equal superposition state over the frequency registers, followed by a Quantum Fourier Transform applied to each coordinate frequency register, yields the same result as the algorithm described previously. We define a regular grid over the inputs $\{\alpha_l\}$ such that $\frac{1}{\sqrt{|\mathcal{L}|}}\sum_{l\in\mathcal{L}} \alpha_l = |+\rangle$

$$|0\rangle\,|0\rangle \xrightarrow{H^{\otimes dm}\otimes I} \sum_{l\in\mathcal{L}} |\alpha_l\rangle \otimes |0\rangle \tag{5.64}$$

$$\xrightarrow{O_U} \sum_{l\in\mathcal{L}} |\alpha_l\rangle \otimes U(\alpha_l)\,|0\rangle \tag{5.65}$$

$$\xrightarrow{\mathrm{QFT}_m^{\otimes d}\otimes I} \sum_{l\in\mathcal{L}}\sum_{k\in[1,n]} a_{l,k}\,|l\rangle\,|k\rangle \ . \tag{5.66}$$

The above result provides an analogue for states decomposition as in Theorem 5.1; similarly, we can obtain the analogue for PQC-functions as in Corollary 5.1. For functions, an amplitude oracle can be defined as follows. For a function $f : \{0,1\}^* \to [0,1]$ that takes a binary decomposition over

α, an amplitude oracle O_f yields:

$$|\alpha\rangle \otimes |0\rangle \xrightarrow{O_f} |\alpha\rangle \otimes \Big(f(\alpha)\,|0\rangle + \sqrt{1-f(\alpha)^2}\,|1\rangle \Big)\,.$$

As in the previous case, we initialize the α register in the equal superposition state, or equivalently in a superposition on the regular grid. Then we apply the oracle and then the QFT on the frequency registers to finally retrieve the Fourier decomposition of f as an amplitude-encoded state.

5.7.6. Proof of the PAC learnability of the Hamiltonian dynamics concept class

Training: We consider quantum circuits realized by using r Trotter steps of the time evolution of the following parameterized Hamiltonian

$$H(x,\alpha) = H'(x) + \sum_{1\le s\le d} \alpha_s P_s\,, \tag{5.67}$$

where P_s are Pauli strings. Such circuits have a frequency space of $m = |\mathcal{L}| = (4r+1)^d$. We implement the circuit learning algorithm exactly as in Section 5.3.2. The difference is that now the labels are off by the Trotter error, that is

$$\varepsilon_y < \frac{t^2 A}{2r}\,, \tag{5.68}$$

where A is the sum of the spectral norm of all pairs of commutators. Using Theorem 5.6 we are guaranteed that the labelling error on a new data point will be smaller ε with probability $1-\delta$ if we choose the following learning parameters

$$T > \frac{16m^4\sqrt{2\log\left(\frac{2m}{\delta}\right)}}{\varepsilon^2},\ \varepsilon_b < \frac{0.2\varepsilon}{m},\ \varepsilon_y < 0.5\varepsilon\,. \tag{5.69}$$

The condition on ε_y yields a requirement on the number of Trotter steps as $r > t^2A/\varepsilon$ which in turn yields the feature space dimension as $m > (4t^2A/\varepsilon+1)^d$. We therefore require the number of training data points to scale as

$$T \in \tilde{\Theta}\left(\frac{\sqrt{d}}{\varepsilon^{4d+2}}\right) \subset \tilde{\Theta}(\mathrm{poly}(n,\varepsilon^{-1},\delta^{-1}))\,, \tag{5.70}$$

so this method is sample-efficient. Regarding computational complexity, the training state requires the execution of $K = mT/\varepsilon_b^2$ poly-depth quantum circuits with the condition of $\varepsilon_b = 0.2\varepsilon/m$. Recalling that $d \in O(\log(n))$,

$$K \in \tilde{\Theta}\left(\frac{\sqrt{d}}{\varepsilon^{7d+4}}\right) \subset \tilde{\Theta}(\mathrm{poly}(n, \varepsilon^{-1}, \delta^{-1})) \,. \tag{5.71}$$

Therefore, the overall training process is efficient on a quantum computer.

Inference: The inference process takes place analogously to the one in Section 5.3.2. Given a new datapoint $x_{t'}$, one retrieves the Fourier coefficients $b_{t'}$ of the approximate Trotter function. This requires the execution of $m/\varepsilon_b^2 \in \tilde{\Theta}(m^3/\varepsilon^2) \subset \tilde{\Theta}(\mathrm{poly}(n, \varepsilon^{-1}, \delta^{-1}))$ poly-depth circuits. Then using the weight vector $\hat{w}$ derived in the training phase, the model returns $y_{t'} = \hat{w} \cdot \hat{b}(x_{t'})$.

5.7.7. PAC efficient Kernel-based algorithm

In this section, we propose a kernel based approach and proves that it efficiently PAC learns the parameterized circuit concept class as in Definition 5.2 if the spectrum has a polynomially large spectrum.

Concept class and noisy data

Consider a U_{hard} that decides a BQP-complete language with the function c_0 defined as follows

$$c_0(x) := \langle x| U_{\text{hard}}^\dagger Z_0 U_{\text{hard}} |x\rangle \tag{5.72}$$

We define the concept class where gates parameterized by an unknown vector $\alpha \in \mathbb{R}^d$ are added to the circuit implementing U_{hard}, such that the concept have a finite Fourier decomposition, as follows

$$c_\alpha(x) = \langle b(x) \mid w(\alpha)\rangle, w(\alpha) = [e^{i\alpha\cdot l}]_{l\in[-L,+L]^d}, b(x), w(\alpha) \in \mathbb{C}^m \,. \tag{5.73}$$

We have access to a dataset $\{(x_t, y_t = c_\alpha(x_t)\}_{t\in[1,T]}$. We define the feature space representation of the data $B = [b(x_t)] \in \mathbb{C}^{T\times m}$, and the Gram matrix $K = [\langle b(x_t) \mid b(x_{t'})\rangle]_{t,t'} \in \mathbb{C}^{T\times T}$. We have access to an approximation of the Gram matrix with $\hat{K} = K + E$ with E p.s.d. and each element is bounded by ε_k, which depends on the number of shots we use to measure the overlap, see Figure 5.8.

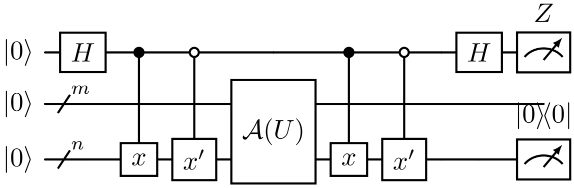


Figure 5.8.: Evaluation of the kernel overlap, we have that $\langle Z \otimes I \otimes |0\rangle\langle 0|\rangle = b(x) \cdot b(x')$

Getting the overlap

We have a circuit such that

$$\mathcal{A}(U)\,|0\rangle\,|x\rangle = |b(x)\rangle\,|x\rangle + |\cdots\rangle\,|x\rangle^{\perp} \tag{5.74}$$

The goal is to find a circuit such that the expectation of an observable yields $\langle b(x)|b(x')\rangle$. We present such a circuit in Figure 5.8, and prove below that it yields the desired outputs. The application of U and then of the last Hadamard gate yields the following states:

$$\frac{1}{\sqrt{2}}(|0\rangle\,|0\rangle\,|x\rangle + |1\rangle\,|0\rangle\,|x'\rangle) \xrightarrow{I\otimes U} \tag{5.75}$$

$$\frac{1}{\sqrt{2}}(|0\rangle\,|b(x)\rangle\,|0\rangle + |1\rangle\,|b(x')\rangle\,|0\rangle + \cdots) \xrightarrow{H\otimes I} \tag{5.76}$$

$$\frac{1}{2}(|0\rangle\,(|b(x)\rangle + |b(x')\rangle)\,|0\rangle + |1\rangle\,(|b(x)\rangle - |b(x')\rangle)\,|0\rangle + \cdots) \tag{5.77}$$

Finally the expectation value of $Z \otimes I \otimes |0\rangle\langle 0|$ (note it has unit spectral norm) is

$$\frac{1}{4}(||b(x)\rangle + |b(x')\rangle|^2 - ||b(x)\rangle - |b(x')\rangle|^2) = \mathrm{Re}(\langle b(x)|b(x')\rangle)\,. \tag{5.78}$$

5.7.8. Noisy Kernel Ridge Regression

In this subsection we derive a bound on the squared prediction error of kernel ridge regression when both the Gram matrix and the labels are observed with noise. For consistency with earlier sections, we temporarily simplify notation and write $x = b(x)$, $w = w(\alpha)$ and $y = c_\alpha(x)$.

Problem setting: We consider the linear model with input $x \in \mathbb{R}^d$

and unknown parameter $w \in \mathbb{R}^d$:

$$y = x^\top w. \tag{5.79}$$

We observe T training samples collected in a data matrix $X \in \mathbb{R}^{T\times d}$, yielding the Gram matrix and label vector

$$K := XX^\top \in \mathbb{R}^{T\times T}, \qquad Y := Xw \in \mathbb{R}^T. \tag{5.80}$$

Instead of (K, Y) we are given noisy observations, where the noisy matrix is potentially corrected to be positive semi-definite [167].

$$\hat{K} = K + E_K, \qquad \hat{Y} = Y + E_Y, \tag{5.81}$$

with entrywise bounds

$$\|E_K\|_\infty \leq \varepsilon_k, \qquad \|E_Y\|_\infty \leq \varepsilon_y. \tag{5.82}$$

For a new test point $x' \in \mathbb{R}^d$ we define the kernel evaluation vector

$$F := Xx' \in \mathbb{R}^T, \tag{5.83}$$

and we observe a noisy version

$$\hat{F} = F + E_F, \qquad \|E_F\|_\infty \leq \varepsilon_k. \tag{5.84}$$

Assume

$$\|w\|_2 \leq B, \qquad k(x, x) \leq \kappa, \qquad \|Y\|_\infty \leq M \tag{5.85}$$

where k is the kernel used by the Kernel Ridge Regression (KRR). Define $\lambda = T\lambda_0 > 0$ and the KRR solution

$$a_{K,Y} = (K + \lambda I)^{-1} Y \tag{5.86}$$

and predict with test evaluations via

$$h_{K,Y}(x') = a_{K,Y} \cdot F \tag{5.87}$$

In reality we only have access to noisy evaluation and get

$$h_{\hat{K},\hat{Y}}(x') = a_{\hat{K},\hat{Y}} \cdot \hat{F} \tag{5.88}$$

Proposition 1 of [168] gives (for exact test evaluation)

$$|h_{\hat{K},Y}(x) - h_{K,Y}(x)| \le \frac{\kappa M}{\lambda_0^2 T} \|\hat{K} - K\|_2 \tag{5.89}$$

and since $\|\hat{K} - K\|_2 \le T\|\hat{K} - K\|_\infty \le T\varepsilon_k$ we obtain the bound

$$|h_{\hat{K},Y}(x) - h_{K,Y}(x)| \le \frac{\kappa M}{\lambda_0^2} \varepsilon_k \tag{5.90}$$

Using $\hat{K} \succeq 0$ so $\|a\|_2 \le \|y\|_2/\lambda \le \sqrt{T}M/(T\lambda_0)$ and $\|a\|_1 \le \sqrt{T}\|a\|_2$, the noisy evaluation for new data contributes

$$|a \cdot (\hat{F} - F)| \le \|a\|_1 \, \varepsilon_k \le \frac{M}{\lambda_0} \varepsilon_k \tag{5.91}$$

Finally, we address the noise on the labels, which impacts the output of the KRR as follows:

$$a_{K,\hat{Y}} - a_{K,Y} = (K + \lambda I)^{-1} E_Y, \qquad h_{K,\hat{Y}}(x') - h_{K,Y}(x') = F^\top (K + \lambda I)^{-1} E_Y. \tag{5.92}$$

Using $\|(K + \lambda I)^{-1}\|_2 \le 1/\lambda$, $\|E_Y\|_2 \le \sqrt{T}\|E_Y\|_\infty \le \sqrt{T}\varepsilon_y$, and $\|F\|_2 \le \sqrt{T}\kappa$ (since $|F_i| = |k(x_i, x')| \le \kappa$), we obtain

$$|h_{K,\hat{Y}}(x') - h_{K,Y}(x')| \le \|F\|_2 \, \|(K + \lambda I)^{-1}\|_2 \, \|E_Y\|_2 \le \frac{\kappa}{\lambda_0} \, \varepsilon_y. \tag{5.93}$$

Combining the effect of all noise sources yields

$$|h_{\hat{K},\hat{Y}}(x') - h_{K,Y}(x')| \le \frac{\kappa M}{\lambda_0^2} \varepsilon_k + \frac{\kappa}{\lambda_0} \varepsilon_y + \frac{M}{\lambda_0} \varepsilon_k. \tag{5.94}$$

Moreover, noiseless KRR with regularization $\lambda = T\lambda_0$ is uniformly stable; hence by [169], for bounded loss, its population risk concentrates around its empirical risk at rate $O(1/T)$, implying PAC learnability, while the additional degradation due to kernel/label noise is controlled by the additive drift bound above polynomial in all relevant quantities.

5.7.9. Conclusion

The proposed kernel-based learning algorithm is able to PAC learn the concept Definition 5.2 under some conditions. Because we have $\kappa = 1$, when B is at most polynomial, we can find parameters $\varepsilon_k, \varepsilon_y$ and T that scale polynomially, guaranteeing the efficiency of the learning algorithm. In particular, if the spectrum is sparse, then there exist a polynomially sized subset of indices $L' = \{l'\}$ such that any x the $b_{l \notin L'}(x) = 0$, then $B \in \mathrm{poly}(n)$.

5.7.10. Flipped concept and connection to RFF

The flipped concept, where the input and the index of the concept class are inverted as, is in contrast to the concept class $\mathcal{C}^U$ we study easy to learn. We define it as follows,

$$\bar{\mathcal{C}} := \{c_x : \alpha \in \mathbb{R}^d \to \sum_l b_l(x) e^{il\cdot\alpha}\}_{x\in\{0,1\}^*} \,.$$

This corresponds to a scenario where a quantum circuit has some fixed, potentially unknown gates and some parameterized gates. Although this concept class looks similar to the one we studied earlier, for this one it is easy to see that unlike $\mathcal{C}^U_{n,\log n}$, the concepts here are in $\mathsf{P/poly}$, where the advice is the polynomially sized list of $b_l(x)$. Therefore, the arguments we make about the hardness of $\mathcal{C}_{n,\log n}$ do not apply.

In fact, this concept is classically efficiently learnable by a simple Fourier analysis of the data. This is typically the scenario of quantum neural networks and quantum kernels, which have been dequantized by techniques like Random Fourier Features [40, 63, 170]. In fact, using the circuit proposed and sampling from the frequency register after post-selection on the circuit register, yields the optimal distribution to approximate the circuit using RFF, as it satisfies perfectly the alignment condition.

Part II.

Simulating classical and quantum systems on bosonic systems

CHAPTER 6

Continuous Variables Quantum Algorithm for solving Ordinary Differential Equations

6.1. Introduction

Differential equations are fundamental to understanding the dynamic behavior of various natural phenomena and technological processes, making them indispensable in a wide range of scientific, engineering, and mathematical disciplines. In the community, the typical problem one studies is the Initial Value Problem (IVP).

Problem 6.1
Given an initial condition $u_0 \in \mathbb{R}^N$*, an analytic function* $\boldsymbol{F} : \mathbb{R}^N \to \mathbb{R}^N$*, and an integration time* $t \in \mathbb{R}_{\geq 0}$ *find the solution* $u \colon \mathbb{R}_{\geq 0} \to \mathbb{R}^N$ *to the differential equation* $\frac{du}{dt} = \boldsymbol{F}(u)$*.*

Many works have tackled this problem, with linearization and with space and time discretization techniques, finding an approximation to the IVP of differential equations can be turned into solving a high-dimensional linear system of equations of the form $Ax = b$. This is the ideal setting

---

The contents of this chapter have been published in [38].

for the application of the HHL algorithm [171] which in some situations promises to solve with an exponential advantage in the size of the system, returning the solution as an amplitude-encoded state $|\mathbf{u}\rangle = \frac{1}{\|u\|}\sum_{j=1}^{N} u_j |j\rangle$. Several works have proposed approaches that scale polylogarithmically in the size of the problem N for linear ODEs [42] and non-linear ODEs [172]. However, it is known that the HHL algorithm is subject to certain limitations [171, 173]: the state preparation of $|b\rangle$, the readout of the results from the amplitude-encoded state $|\mathbf{u}\rangle$, the sparsity of A, and the condition number of A. In particular, related to that last caveat, a lower bound of the complexity has been proven in [174] that applies to any quantum algorithm returning the solution of an ODE as an amplitude-encoded state, which is the case for the vast majority of exact quantum ODE solvers. Their complexity is bounded by $\Omega(e^{\delta t})$ where t is the integration time of the IVP and δ is a parameter derived from the spectrum of the matrix M characterizing the linear ODE $\dot{x} = Mx$.

In this chapter we propose a different approach to solve ODEs, and explore to which extent the above limitations could be circumvented. We study the translation of arbitrary dynamics into a Schrodinger-like equation [175]. Mapping the native space of the ODE to a space where the time evolution is unitary effectively reduces an arbitrary ODE problem to a Hamiltonian time evolution. While previous works approximate the infinite-dimensional Hilbert space with a finite-dimensional Hilbert space, in this chapter, we will explore an alternative way of dealing with the infinite dimensionality of the Hilbert space, working directly in a continuous variable framework.

The rest of the chapter is structured as follows: in Section 6.2 we cover background information such as the Koopman–von Neumann (KvN) framework and how previous works have used this formalism in quantum algorithms for ODEs. We also introduce continuous variable computing. In Section 6.3, we will present our main contribution, a continuous variable algorithm making use of the KvN framework, and explain how it is better suited to solve the initial distribution problem rather than the initial value problem. In Section 6.4 we discuss the limitations of the proposed algorithm and the next steps. Finally, we present our conclusions in Section 6.5.

6.2. Background

6.2.1. Koopman–von Neumann classical mechanics

The Koopman–von Neumann classical mechanics describe the evolution of arbitrary classical dynamical systems as the Hamiltonian evolution of wave functions in an infinite-dimensional Hilbert space, where each mode corresponds to one of the N coordinates of the ODE. While the full theory behind this framework is more comprehensive, we present the basic elements of the Koopman–von Neumann (KvN) framework [176]. The central element of this framework is the Hamiltonian describing the time evolution. For an ODE defined as $\frac{du}{dt} = \mathbf{F}(u)$, we define the KvN Hamiltonian as

$$H_{\mathbf{F}} := \frac{1}{2} \left(\hat{p}\mathbf{F}(\hat{q}) + \mathbf{F}(\hat{q})\hat{p} \right) \tag{6.1}$$

where $\hat{q} = [\hat{q}_1, \hat{q}_2, ..., \hat{q}_N]$ is the vector of the position operators for each of the N modes, and $\hat{p}$ is the vector of their respective momentum operators. We also use the notation $|x\rangle_{\hat{q}}$ for a position operator eigenstate with eigenvalue x. Looking at the position operator in the Heisenberg picture, we can write that the operator follows the same equation as the ODE $\dot{x} = \mathbf{F}(x)$ (full derivation in [177]):

$$\frac{d\hat{q}}{dt} = \frac{[\hat{q}, H]}{\mathbf{i}h} = \mathbf{F}(\hat{q}). \tag{6.2}$$

Therefore, the expectation value of the position operator follows the trajectory of the ODE. Applying a position eigenstate $|u_0\rangle_{\hat{q}}$ to Equation (6.2), we can show that KvN evolution of the position eigenstate with an eigenvalue equal to the initial condition will result in a position eigenstate with the corresponding eigenvalue following exactly the solution of the ODE

$$|u(t)\rangle_{\hat{q}} = e^{iH_{\mathbf{F}}t} |u_0\rangle_{\hat{q}}, \tag{6.3}$$

where $u(t)$ is the solution to the IVP (Problem 6.1). As explained in [178] the state of a system in an infinite-dimensional Hilbert space is described by a wave function that can be expressed in the $\hat{q}$ representation $|\psi\rangle = \int \psi(u) |u\rangle_{\hat{q}} du$. Thanks to the linearity of the time evolution, evolving a wave function described by $\psi : u \to \sqrt{p_0(u)}$ using the KvN Hamiltonian will correspond to evolving the initial distribution $p_0 \colon \mathbb{R}^n \to [0, 1]$, solving a slightly different problem to the IVP. We introduce the initial distribution problem (IDP) which is, as we will see, a better fit for the types of computations we can expect CV quantum computers to perform. It is also a natural extension of the IVP when there is uncertainty in the initial state.

Problem 6.2
Given an initial probability distribution $p_0 \colon \mathbb{R}^N \to [0,1]$*, an analytic function* $\boldsymbol{F} : \mathbb{R}^N \to \mathbb{R}^N$*, and an integration time* $t \in \mathbb{R}_{\geq 0}$ *find the probability distribution* $p_t \colon \mathbb{R}^N \to [0,1]$ *evolved for a time* t *according to the differential equation* $\frac{du}{dt} = \boldsymbol{F}(u)$*.*

6.2.2. Previous work: Finite Hilbert space approximations

The use of the KvN framework in the context of solving the IVP has rapidly been identified as an opportunity for quantum computing [176, 179, 180]. To the best of our knowledge, all previous approaches reduce the infinite-dimensional Hilbert space to a finite dimension to perform an approximate simulation on a qubit-based quantum computer. This is done either with a truncation of the infinite-dimensional Hilbert space or via discretization of the phase space. However, the reduction to a finite Hilbert space is far from trivial [181], as finding rigorous bounds for the truncation error is still a work in progress. We explore an alternative way by working with infinite dimensional systems, which then avoids truncation problems, but does so at a cost. This introduces another class of issues, such as the appearance of the norm of unbounded operators in the Trotter error. Specifically, we propose an algorithm directly compiled as a sequence of continuous variables gates, meant to be executed on a bosonic quantum computer (e.g. photonics).

6.3. Proposed Algorithm

6.3.1. Approximation of the time evolution of the KvN Hamiltonian

Using the generators corresponding to the set of gates defined in Section 2.4.1, in this section we propose an algorithm to derive a sequence of gates approximating the time evolution of the Koopman–von Neumann Hamiltonian for a one-dimensional polynomial ODE. Any polynomial function describing the ODE can be written as $\mathbf{F}(u) = \sum_{k=0}^{d} a_k u^k$, and the corresponding KvN Hamiltonian is $H = \sum_k^d a_k \frac{1}{2}\{\hat{p}, \hat{q}^k\}$, where $\{A, B\} = (AB + BA)$ is the anti-commutator. Because of the way this

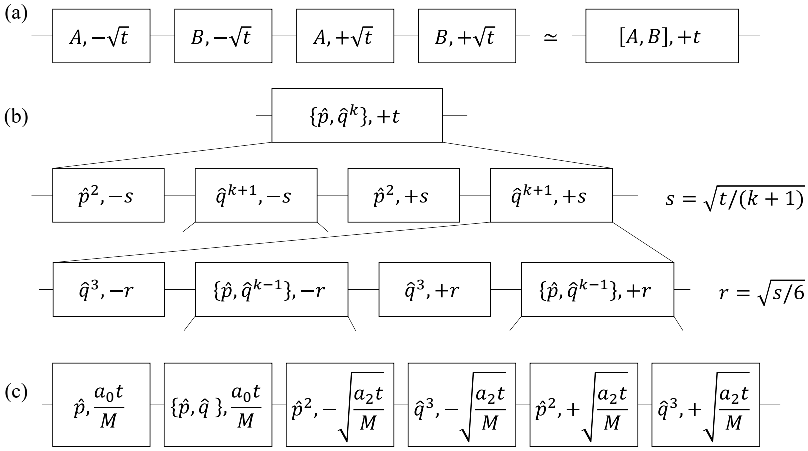

(a)
$A,-\sqrt{t}$
$B,-\sqrt{t}$
$A,+\sqrt{t}$
$B,+\sqrt{t}$
$\simeq$
$[A,B],+t$
(b)
$\{\hat{p},\hat{q}^k\},+t$
$\hat{p}^2,-s$
$\hat{q}^{k+1},-s$
$\hat{p}^2,+s$
$\hat{q}^{k+1},+s$
$s=\sqrt{t/(k+1)}$
$\hat{q}^3,-r$
$\{\hat{p},\hat{q}^{k-1}\},-r$
$\hat{q}^3,+r$
$\{\hat{p},\hat{q}^{k-1}\},+r$
$r=\sqrt{s/6}$
(c)
$\hat{p},\frac{a_0 t}{M}$
$\{\hat{p},\hat{q}\},\frac{a_0 t}{M}$
$\hat{p}^2,-\sqrt{\frac{a_2 t}{M}}$
$\hat{q}^3,-\sqrt{\frac{a_2 t}{M}}$
$\hat{p}^2,+\sqrt{\frac{a_2 t}{M}}$
$\hat{q}^3,+\sqrt{\frac{a_2 t}{M}}$

Figure 6.1.: (a) Group commutation relationship used to approximate a gate, (b) two-step nested structure to expand any monomial gate as per Algorithm 2. This leads to a recursive decomposition that is stopped when $\hat{q}^{k+1}=\hat{q}^3$ in which case the native gate is used. (c) this sequence repeated M times is an example of a simple Trotter scheme for a quadratic 1D ODE

Hamiltonian naturally decomposes as a sum of Hamiltonian that are close to the native set of gates, we choose a Trotterisation [182] approach to evolve this Hamiltonian. We break down the algorithm into two parts, each of which is detailed in the Appendix in Section 6.6.1. Firstly, we introduce the monomial subroutine (Algorithm 2) to approximate any gate with a generator of the form $\{\hat{p},\hat{q}^k\}\forall k\in\mathbb{N}$. We use the group commutation relationship depicted in Figure 6.1-(a) in a nested recursive expansion as illustrated in Figure 6.1-(b). Secondly, we use the monomial subroutine to be able to approximate any polynomial as detailed in Algorithm 3, and in particular the polynomial of the Hamiltonian of interest H.

Computing the error for such an approximation is a challenge, as it would require a complete Trotter error theory [182] for continuous variables which, to the best of our knowledge, has not yet been resolved. Naive extensions from finite systems run into trouble as the operators become unbounded.

Indeed, the general Trotter error is expressed as $O(\|[A,B]\|t^2)$, and as in infinite-dimensional Hilbert spaces operator norms are not bounded, this would, in theory, yield an infinite error. We present in the Appendix a potential direction to deal with this.

As an example, we present a simple scheme for an arbitrary 1D quadratic ODE in Figure 6.1-(c). This sequence of gates is derived by applying Algorithm 3 to a generic quadratic polynomial $\mathbf{F}(u) = a_0 + a_1 u + a_2 u^2$. The associated error comes in two different scalings. On the one hand, the first two blocks correspond to the linear part of the ODE $(a_0 + a_1 u)$, their time parameter evolving linearly with time. Therefore general knowledge of Trotter theory yields an error $O((t/M)^2)$. On the other hand, the four last blocks correspond to the quadratic part of the ODE $(a_2 u^2)$, and applying the Baker–Campbell–Hausdorff formula to Figure 6.1-(a) the error scales as $O((t/M)^{3/2+1})$. Comparing these two error scalings we realize that one is smaller than the other when ideally they should be of the same order. This illustrates that the above scheme could benefit from a better sequence of numbers of trotter steps M (having fewer linear blocks than quadratic blocks for example). Finally, for the overall repetition M times, the error ε will be $O((t/M)^{5/2})$ and the number of cubic gates required for the quadratic scheme is $2M = O(\varepsilon^{2/5} t)$ with a potentially large prefactor due to the commutator norms, to be determined in future work.

6.3.2. Chaos and position eigenstates

The theoretical framework presented in Section 6.2.1 uses position eigenstates. However, these states are not physical states, as the amplitude of the wave function expressed in the Fock space is not square integrable. They can be interpreted as infinitely squeezed states, therefore we decided to approximate position eigenstates with finitely squeezed states. This approximation is equivalent to approximating the Dirac delta with a Gaussian where the standard deviation is controllable. A state squeezed by an amount r, (meaning that the squeezing generator is time evolved for a time r starting from a coherent state), yields a standard deviation of $\sigma_0 = e^{-r}$. Then displacing this state to the position x_0 can be written as

$$|x_0, \sigma_0\rangle = \frac{1}{\sigma_0\sqrt{\pi}} \int_{-\infty}^{+\infty} \exp\left(-\frac{(x-x_0)^2}{\sigma_0^2}\right) |x\rangle_{\hat{q}}\, dx. \tag{6.4}$$

Using the position eigenstates basis for decomposition, we realize that this is a continuous weighted sum over position eigenstates and because the time evolution is linear, the time evolution of the sum of position eigenstates is equal to the sum of the time evolution of position eigenstates. This means that effectively the resulting state will correspond to the evolution of a Gaussian distribution as the initial distribution under the differential equation.

Therefore in order to solve the IVP, we need to find a bound for the variance of the solution. Using knowledge of an upper bound of the Lyapunov exponent λ, with a required accuracy ε and an integration time t, we can use it to choose the initial squeezing factor of the state. Indeed per the definition of the Lyapunov exponent, the standard deviation of the final state would be upper bounded: $\sigma_f < \sigma_0 e^{\lambda t}$ therefore in order to have the requested accuracy ε we need to have $\varepsilon < e^{-r} e^{\lambda t}$, and we can conclude we need the squeezing factor to be

$$r > \lambda t + \log(\varepsilon). \tag{6.5}$$

It is important to note that this is not an unbiased estimator. For non-linear ODEs it is not guaranteed that the mean of the evolution of a Gaussian distribution will evolve accordingly to the solution of the ODE with a Dirac delta as an initial condition. Our suggestion to remedy that is to sufficiently squeeze the state initially.

However, we propose to use the algorithm to solve the IDP instead of the IVP, focusing on the evolution of the distribution itself rather than a single initial condition. Executing the algorithm on a continuous variables quantum computer becomes a machine to sample from that evolved distribution. This could for example help identify attractors or other properties of the dynamical system. Such properties include for example the Lyapunov exponent, which could be derived as the variance of the position operator $\hat{q}$.

6.3.3. Overall algorithm

We present a description of the algorithm for a 1D polynomial ODE in Algorithm 1, and explain it here. Starting with the preparation of the state, the vacuum state is first squeezed according to the required precision. Each mode is then displaced by an amount corresponding to the initial value. Secondly, we implement the approximated version of the time evolution of the KvN Hamiltonian, by executing the sequence of gates returned as in Section 6.3.1. Finally, we perform a homodyne measurement of the position and return the sampled value.

```
Algorithm 1 Solve 1D-Poly-ODE
  Inputs :
    • $\mathbf{F}(u) = \sum_{k=0}^{d} a_k u^k$ the polynomial function characterising the ODE
    • $u_0 = u(0) \in \mathbb{R}$ the initial condition
    • $t$ the requested time interval
    • An $M$ sequence (designed to reach an accuracy $\varepsilon$)
    • $\lambda$ an upper bound on the Lyapunov exponent
  Outputs : $\tilde{u}(t)$ an approximation of the ODE solution $u(t)$ such that $\|\tilde{u}(t) - u(t)\| < \varepsilon$
  Execute
  start from the vacuum state $|0\rangle$
  $\{\hat{p}, \hat{q}\}$ , $\lambda t + \log(\varepsilon)$
  $\hat{p}$, $u_0$                                          ▷ Initial state preparation
  apply the sequence G from Algorithm 3 Poly$(d, \{a_k\}_{k \in [0,d]}, t)$   ▷ Time evolution
  sample and return $\langle \psi \mid \hat{q} \mid \psi \rangle$            ▷ Homodyne measurement
```

6.4. Discussion and next steps

So far we have looked into a single-mode algorithm, but adding beam-splitters guarantees the full universality of CV for several modes. The explicit procedure to extend the algorithm proposed in Section 6.3 from one-dimensional ODEs to multi-dimensional ODEs is out of scope of this chapter but will be considered in future work. It is however worth noting that beam-splitters can be considered a cheap gate in photonics.

As discussed in Section 6.3.1 and in the Appendix, more work is required to fully characterize the Trotter error in infinite-dimensional Hilbert spaces. Such analysis will enable choosing the appropriate number of Trotter steps M at each commutator breakdown (Figure 6.1-(a)). We highlight the difference with the truncated approaches that limit support of the state which shall be a finite number of Fock states, while the Trotter approach limits the expected energy of the state but enables infinite-dimensional support. For this reason, we expect the final error scaling to be different between our approach and the previously introduced methods relying on Hilbert space truncation.

In addition, the difficult choice of the number of Trotter steps at each commutator breakdown hints towards a potential avenue performing a variational version of this algorithm, with sequences of displacement, squeezing, quadratic and cubic gates with tuneable execution time parameters. These parameters would be optimized to maximize the match between the output of the algorithm and given time-series data.

Finally, with fully characterized tighter Trotter error bounds, the main goal is to try and compare the complexity of such an algorithm to that of qubit-based algorithms. For example the Stellar rank [183] corresponds to the number of zeros of the Husimi function (a phase space representation of a state) that characterizes the "non-Gaussianity" of a state. The stellar rank is equivalent to the minimal number of photon additions necessary to engineer the state. This enables the introduction of an infinite hierarchy of states, that can be used in the context of the study of complexity of continuous variables quantum computing.

6.5. Conclusion

We propose a new perspective on the Koopman–von Neumann formalism, that allows mapping arbitrary classical dynamics to an infinite-dimensional Hilbert space where the dynamics are unitary. This effectively transforms an ODE problem into a Hamiltonian evolution problem. While previous works consider these dynamics in an approximated finite Hilbert space, we chose to keep working in an infinite-dimensional Hilbert space and propose a Continuous Variable Algorithm to solve one-dimensional polynomial ODEs. We propose that for such a CV system, the natural problem to be tackled is not the basic IVP, but rather its natural probabilistic extension the IDP. The current work does not allow for a full complexity analysis due to the problems with the estimation of the Trotter error, however, we do provide the algorithm with a sub-optimal sequence solving an arbitrary quadratic ODE, including the scaling of the number of gates.

6.6. Appendix

6.6.1. Algorithm subroutines

For the first subroutine, we will use the following commutation formulas extensively:

$$\left[\hat{p}^2, \hat{q}^{k+1}\right] = (k+1)i\{\hat{p}, \hat{q}^k\} \tag{6.6}$$

$$\left[\{\hat{p}, \hat{q}^k - 1\}, \hat{q}^3\right] = 6iq^{k+1} \tag{6.7}$$

The first subroutine's goal is to approximate a monomial of quadrature commutators of arbitrary degree $k : \{\hat{p}, \hat{q}^k\}$. The degree $k = 0$ simply corresponds to the generator of the position displacement gate $H = \{\hat{p}, \hat{q}^0\} = 2\hat{p}$. The degree $k = 1$ corresponds to the generator of the squeezing gate $H = \{\hat{p}, \hat{q}^1\} = 2(\hat{p}\hat{q} + \hat{q}\hat{p})$. For higher degrees, generators of the form $\{\hat{p}, \hat{q}^k\}$ are not part of our native set of gates as described in Section 2.4.1. From Equation (6.6) we can obtain it as the commutator between $\hat{p}^2$ and $\hat{q}^{k+1}$ using the following approximation derived from second-order Trotter $e^{-ihA}e^{-ihB}e^{+ihA}e^{+ihB} = e^{-h^2[A,B]+O(h^3)}$. However, for $k \leq 3$ $\hat{q}^{k+1}$ is out of our set of gates. Looking at Equation (6.7) we realise that $\hat{q}^{k+1}$ is proportional to the commutator between $\{\hat{p}, \hat{q}^k - 1\}$ and $\hat{q}^3$. These principles are illustrated in Figure 6.1-(b), and we present the recursive Algorithm 2 to generate a sequence of gates that approximate a monomial of arbitrary degree. Sequences of gates are presented as a sequence of generators evolved for a certain time, using the following operators: the $+$ operation concatenates lists, the $*$ operation with an integer M repeats M times the sequence, $-$ is a sequence realized backward with opposite times (e.g. $-[\ \{A, t\}, \{B, -s\}] = [\ \{B, s\}, \{A, -t\}]$. Using this Algorithm 2 we can generate the gate sequence for the time evolution of the full KvN Hamiltonian $H = \sum_k^d a_k \{\hat{p}, \hat{q}^k\}$ Trotterising each of the monomials, as detailed in the Algorithm 3.

6.6.2. Trotter error and unbounded operators

The general Trotter error is expressed as $O(\|[A, B]\| t^2)$, which includes the spectral norm of an unbounded operator. However, on physical computers the energy of states is limited, therefore we present a potential way forward introducing the energy-constrained circle norm (ECCN), inspired by the energy-constrained diamond norm that is extensively used in the study of continuous variable channel capacities [184]. We define the energy-constrained circle norm for an operator M, and using $\hat{N}$ for the number operator as:

$$\|M\|_N := \max_{|\psi\rangle, \langle\psi|\hat{N}|\psi\rangle \leq N} \frac{\langle\psi \mid M \mid \psi\rangle}{\langle\psi \mid \psi\rangle} \tag{6.8}$$

For future work, we would on the one hand characterize the ECCN for all gates in our model of computation as seen in Section 2.4.1 as a function of the evolution time and the energy constraint, and on the other hand estimate the evolution of expectation value of the number of photons using the Heisenberg picture $[H, \hat{N}]$. With this combined information we expect it should be possible to find some bounds. This represents an extensive

Algorithm 2 Mono: Recursive Gate sequence to approximate the evolution of $\{\hat{p}, \hat{q}^k\}$

```
Hyper-parameters :
  • $M$ a sequence of integers for the number of Trotter steps for each order.
Inputs :
  • $k \geq 2$ the target degree of the monomial
  • $t$ the requested time interval
Outputs :
  • $G = [\{H_i, t_i\}]_{i\in[0,L]}$ a sequence of gates (defined as a generator for a certain time)
Execute :
if $k = 2$ then                                                        ▷ end of recursive
    $s := \frac{1}{M(k,1)}\sqrt{\frac{t}{3}}$
    $G := [\{\hat{p}^2, -s\}, \{\hat{q}^3, -s\}], \{\hat{p}^2, +s\}], \{\hat{q}^3, +s\}]] * M(k,1)$
else                                                                   ▷ intermediate levels
    $r := \frac{1}{M(k,1)}\sqrt{\frac{s}{6}}$, to get $\hat{q}^{k+1}$ for $s$ using Equation (6.7):
    $F := \big(\mathbf{Mono}(k-1,-r) + [\hat{q}^3, -r] + \mathbf{Mono}(k-1,+r) + [\hat{q}^3, +r]\big) * M(k,1)$    ▷ recursive
    $s := \frac{1}{M(k,2)}\sqrt{\frac{t}{k+1}}$, to get $\{\hat{p}, \hat{q}^k\}$ for $t$ using Equation (6.6):
    $G := \big(F + [p^2, -s] - F + [p^2, +s]\big) * M(k,2)$
end if
Return $G$
```

Algorithm 3 Poly: approximate time evolution of KvN Hamiltonian

```
Hyper-parameters :
    • $M$ a sequence of integers for the number of Trotter steps for each order.
Inputs :
    • $d$ the degree of the polynomial
    • $\{a_k\}_{k\in[0,d]}$ the coefficient of the polynomial
    • $t$ the requested time interval
Outputs :
    • $G = [\{H_i, t_i\}]_{i\in[0,L]}$ a sequence of gates
Execute :
$G \coloneqq \left[\{\hat{p}, \frac{a_0 t}{M_r M(0)}\}\right] * M(0,0)$
$G \coloneqq G + \left[\{\hat{p}\hat{q} + \hat{q}\hat{p}, \frac{a_1 t}{M_r M(1)}\}\right] * M(1,0)$
for $k = 2$ to $d$ do
    $G \coloneqq G + \mathbf{Mono}\left(k, \frac{a_k t}{M_r M(k,0)}\right) * M(k,0)$        ▷ using Algorithm 2
end for
$G \coloneqq G * M_r$
Return $G$
```

piece of work and is left for later analysis, this chapter's scope being limited to outlining an idea. Such an analysis will enable the optimization of the number of Trotter steps for each breakdown into commutator in the nested structure, tagged M in the Algorithm 2.

CHAPTER 7

The complexity of simulating exponentially large Gaussian bosonic circuits

7.1. Introduction

The study of the power of quantum computers has been a central and fundamental topic in complexity theory. While it is known that these devices can solve problems at least as fast as classical probabilistic computers, we also know that they are unable to do so more than exponentially faster [31, 185]. This has raised the question: *What are the tasks that saturate this separation? I.e., what are the problems for which a quantum computer can achieve an exponential advantage over its classical counterparts?*

To answer such questions one starts with the class Bounded-Error Quantum Polynomial (BQP), the set of decision problems that a quantum computer can solve in polynomial time with a small constant probability of failure. Then, one determines the subset of problems that are the hardest therein, known as BQP-complete. Several BQP-complete problems are known, such as those based on the Harrow-Hassidim-Lloyd algorithm [171],

---

The contents of this chapter have been published in [39].

scattering in scalar quantum field theory [186], and more recently on the quantum simulation of exponentially many coupled classical oscillators [42]. The latter presents the intriguing perspective that simulating exponentially large linear and energy-preserving simple classical systems leads to BQP-completeness, and, under reasonable complexity theory assumptions, to an exponential quantum advantage over classical methods.

In this chapter we prove an analogous result to that in Ref. [42], namely, that the quantum simulation of particle-preserving GB circuits (i.e. passive linear optics) [187–189] acting on Gaussian initial states on exponentially many modes also leads to BQP-completeness (see Figure 7.1). At its core, our framework starts with the realization that a direct simulation of a bosonic system is intractable on a qubit-based quantum computer, as the associated Hilbert spaces are fundamentally different (with one being infinite-dimensional, and the other discrete). This issue can be avoided by restricting the simulation to the first and second moments of the quadrature operators. That is, we encode in a quantum state the position and momentum expectation values (and their covariance matrix) over the initial bosonic state. As such, instead of simulating the action of the GB circuit on the bosonic Hilbert space, we implement its effective action on the expectation values on a gate-based quantum computer.

In this context, we present a constructive dictionary that translates back-and-forth between the symplectic propagator associated with a universal set of GB gates (beamsplitters, phase and squeezing gates) and qubit circuits. The efficiency of our simulation framework relies on several key conditions, such as the input qubit-state being preparable in polynomial time, and the quantum circuit requiring only polynomially-many gates. Indeed, we present cases of interest for which these two conditions are satisfied, and therefore for which we can achieve an exponential quantum advantage.

Finally, we show that adding a layer of squeezing gates to interferometers boosts the complexity of the problem to PostBQP-complete, which is a very powerful complexity class [34] that, among others, contains QMA (Quantum Merlin-Arthur). In PostBQP, in addition to polynomially-sized quantum circuits one obtains the ability to post-select on measurement outcomes, that is, to exponentially amplify small probabilities.

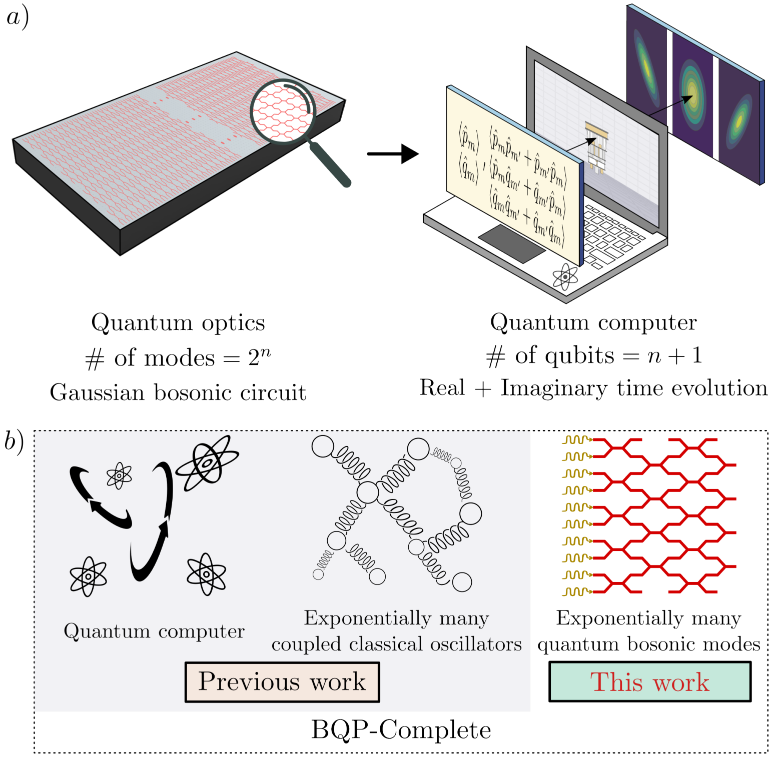

a)
$\langle \hat{p}_m \rangle$, $\langle \hat{q}_m \rangle$
$\langle \hat{p}_m \hat{p}_{m'} + \hat{p}_{m'} \hat{p}_m \rangle$
$\langle \hat{p}_m \hat{q}_{m'} + \hat{q}_{m'} \hat{p}_m \rangle$
$\langle \hat{q}_m \hat{q}_{m'} + \hat{q}_{m'} \hat{q}_m \rangle$
Quantum optics
# of modes = 2^n
Gaussian bosonic circuit
Quantum computer
# of qubits = $n + 1$
Real + Imaginary time evolution
b)
Quantum computer
Exponentially many coupled classical oscillators
Exponentially many quantum bosonic modes
Previous work
This work
BQP-Complete

Figure 7.1.: **Schematic representation of our main results.** a) We present a framework for simulating the action of a GB circuit on the first and second moments of quadrature operators of a bosonic state on 2^n modes on an $(n+1)$-qubit gate-based quantum computer. b) We show that particle-preserving GB evolutions on Gaussian bosonic states are sufficient to define a problem that is BQP-complete, thus indicating that passive linear optics on exponentially many bosonic modes are as powerful as universal quantum computers.

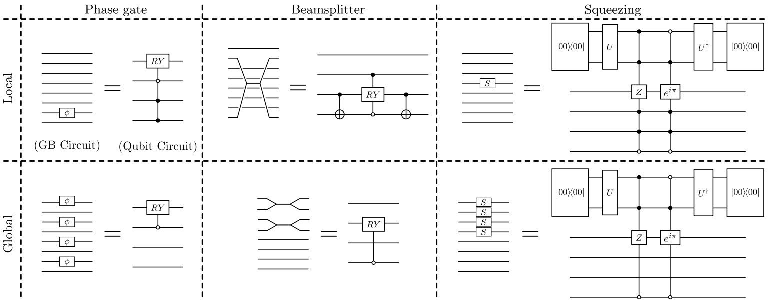

Phase gate
Beamsplitter
Squeezing
Local
Global
(GB Circuit)
(Qubit Circuit)
ϕ
RY
S
$|00\rangle\langle 00|$
U
Z
$e^{i\pi}$
$U^\dagger$

Figure 7.2.: **Examples of GB gates in the qubit picture.** We consider a bosonic system on $M = 8$ modes, leading to a circuit on 4 qubits. The local phase gate acts on the mode $m = 6 = 2^0 \times 0 + 2^1 \times 1 + 2^2 \times 1$. The local beamsplitter acts on the modes $m = 1$ and $m' = 7$, whose binary representations only share the least significant bit. The local squeezing gate acts on mode $m = 3 = 2^0 \times 1 + 2^1 \times 1 + 2^2 \times 0$, and is represented by an imaginary time evolution as a linear combination of unitaries with post-selection on two ancillary qubits (which have been added on top). The global phase gate acts on all modes whose index is even. The global beamsplitter is applied to the first half of the modes, pairing each mode with even index m to its nearest neighbor mode with index $m' = m + 1$. The global squeezing gate is applied to the first half of the modes.

7.2. Gate-Based simulation of Gaussian circuits

7.2.1. Initialization

To start the simulation on the quantum computer, we encode the normalized vector $\langle\hat{\vec{z}}\rangle$ (normalized matrix $\vec{\sigma}$) into a pure (mixed) $(n+1)$-qubit quantum state $\left|\hat{\vec{z}}\right\rangle \propto \langle\hat{\vec{z}}\rangle$ ($\varrho_{\vec{\sigma}} \propto \vec{\sigma}$). For instance, we have

$$\left|\hat{\vec{z}}\right\rangle = \frac{1}{\|\langle\hat{\vec{z}}\rangle\|_2} \sum_{m=1}^{2^n} \langle\hat{q}_m\rangle \left|0\right\rangle \otimes \left|m\right\rangle + \langle\hat{p}_m\rangle \left|1\right\rangle \otimes \left|m\right\rangle \, . \tag{7.1}$$

For our scheme to be efficient, such states need to be preparable in polynomial time. Such cases occur, e.g., when there are $\mathcal{O}(\mathrm{poly}(n))$ non-zero known position and momentum expectation values. Equation (7.1) illustrates the privileged role of the first qubit [190], separating the Hilbert space into two subspaces, one associated with the positions and the other with the momenta. We will henceforth refer to the first qubit as the *symplectic qubit.* The remaining n qubits serve as a register for each of the 2^n modes, and we will refer to them as *register qubits.* In particular, in Equation (7.1) each mode label m is encoded via its binary decomposition as $m = 2^0 m_1 + 2^1 m_2 + \ldots + 2^{n-1} m_n$, with $m_l \in \{0,1\}$, and with the least significant qubit being the top-most register qubit.

Here we note that when ρ_0 is Gaussian, then it is fully characterized by the first and second moments of the quadrature operators, meaning that $\left|\hat{\vec{z}}\right\rangle$ and $\varrho_{\vec{\sigma}}$ provide a full description of these states. Moreover, if the initial state is also coherent (eigenstates of $\hat{a}$), then $\varrho_{\vec{\sigma}}$ is the maximally mixed state on $(n+1)$ qubits. For other non-Gaussian states our framework is restricted to the information contained in $\hat{\vec{z}}$ and $\vec{\sigma}$.

7.2.2. Evolution

Once the initial state is prepared, we can evolve it with a quantum circuit that effectively implements the symplectic propagators associated with the gates in the GB circuit (see for instance Equation (2.39)). In particular, we present a dictionary to efficiently translate between standard GB and qubit gates. We also refer the reader to Figure 7.2 for an explicit circuit depiction of some GB gates in qubit circuit form.

The *phase gate* is a particle-preserving gate acting on a single mode

m. In terms of bosonic operators, its generator is $\hat{H} = \hat{p}_m^2 + \hat{q}_m^2$, yielding $\Omega K = 2iY \otimes |m\rangle\langle m|$ in the qubit picture. This corresponds to an R_y gate (rotation about the y-axis) on the symplectic qubit, conditioned on the $|m\rangle\langle m|$ state on the register.

The *beamsplitter* is a particle-preserving gate acting on two modes m and m'. It is generated by $\hat{H} = \hat{q}_m \hat{p}_{m'} - \hat{q}_{m'} \hat{p}_m$ (with $m \neq m'$), which results in $\Omega K = 2iI \otimes (i\,|m\rangle\langle m'| - i\,|m'\rangle\langle m|)$. This corresponds to an R_y rotation in the subspace spanned by $|m\rangle$ and $|m'\rangle$. This gate can be implemented by using controlled-not gates to transform $|m\rangle$ and $|m'\rangle$ into two computational-basis states that only differ in one qubit, then performing a controlled-R_y rotation on that qubit (conditioned on the other $n-1$ qubits in the register), and applying the same controlled-not gates to return to the original basis. We highlight the fact that this gate acts trivially on the symplectic qubit, as beamsplitters conserve total momentum and total position.

The *squeezing gate* is a non-particle-preserving gate that acts on a single mode m. Its generator is $\hat{H} = \pm(\hat{q}_m \hat{p}_m + \hat{p}_m \hat{q}_m)$, leading to $\Omega K = \pm 2X \otimes |m\rangle\langle m|$. This produces an imaginary-time evolution [94, 191–193], which can be implemented, e.g., with two ancillary qubits and post-selection. As illustrated in Figure 7.2 and further discussed in the SI, we propose to implement it as a linear combination of unitaries [194], with the latter being multi- controlled-Z and controlled-phase gates. Notably, the fact that states cannot be arbitrarily squeezed translates in our framework as the success probability of the imaginary-time evolution for infinite time being zero.

Finally, the *displacement gate*, a common non-particle-preserving Gaussian gate, cannot be included as a linear qubit gate in the proposed framework (see the SI for an additional discussion on this gate).

We stress that while our framework requires the implementation of multi-controlled qubit operations to simulate the action of some GB gates, these can be compiled exactly using only $\mathcal{O}(n)$ local gates [43]. Moreover, if instead of one or two modes, we consider GB gates that act on several (potentially exponentially many) modes, there can be simplifications that render them much easier to implement at the qubit level. As an important example, we show in Figure 7.2 (see also SI) how some global phase gates and beamsplitters acting on 2^{n-1} modes simplify to two-qubit controlled-R_y rotations, as well as how the number of control qubits is reduced for some global squeezing gates. This means that when the GB circuit is composed of polynomially many such local particle-preserving gates, the implementation of the quantum circuit is efficient. When non-particle-preserving squeezing gates are added, then the efficiency will ultimately depend on how many such gates are added, as well as on their squeezing strength parameters.

At this point, we recall that the covariance matrix of coherent states leads to a maximally mixed state $\varrho_{\vec{\sigma}}$, and therefore it remains invariant when evolved with a purely unitary circuit (e.g., particle-preserving GB gates in an interferometer). This result is well aligned with the bosonic picture where coherent states going through interferometers remain coherent states. Then, as squeezing gates are not particle-preserving, their action on a coherent state corresponds to purifying the associated $\varrho_{\vec{\sigma}}$.

7.2.3. Measurements

At the output of the quantum circuit, we obtain states that represent the evolved expectation values and covariance matrix of quadrature operators (which we respectively denote as $\left|\hat{\vec{z}}'\right\rangle$ and $\varrho_{\vec{\sigma}'}$). We now discuss how measuring these states allows us to extract useful information about the GB circuit.

There are a variety of measurements of interest for Gaussian states. As detailed in the SI, photon counting can be implemented for coherent states by sampling bitstrings from $\left|\hat{\vec{z}}'\right\rangle$. To understand why this is the case, we recall that the energy for each mode is proportional to the probability of sampling the corresponding bitstring. Secondly, for any bosonic state homodyne measurements correspond to retrieving the position (momentum) of a specific mode. At the qubit level, these measurements can be implemented from a swap-Hadamard test between $\left|\hat{\vec{z}}'\right\rangle$ and some computational-basis state of interest. Then, our framework also allows us to estimate the fraction of total momentum and total position, as this value can be retrieved by measuring the symplectic qubit in $\left|\hat{\vec{z}}'\right\rangle$. Similarly, the fractional energy of the first and second half of the modes can be estimated by measuring the bottom-most (most significant) register qubit. Moreover, we note that combining computational basis measurements (or Hadamard tests) on $\left|\hat{\vec{z}}'\right\rangle$ and $\varrho_{\vec{\sigma}'}$ allows us to estimate the energy in a given mode. To finish, we note that while the total energy remains constant for particle-preserving GB circuits, this quantity can change when non-particle-preserving gates are included. In this case, while we cannot directly measure the total energy of the system (as the states need to be normalized), one can keep track of the total energy by using as a proxy the success probability of the imaginary-time evolutions.

7.3. BQP-completeness of Exponentially large interferometers

7.3.1. Problem definition

We now show that we can leverage our framework to devise a decision problem based on a restricted class of large optical quantum interferometers and prove that it is BQP-complete. We begin by introducing a family of quantum interferometers, that we refer to as *bit-structured*.

Definition 7.1 (Bit-structured interferometer)
A bit-structured interferometer acting on 2^n nodes consists of L global beamsplitters, such that each global beamsplitter acts on 2^{n-1} modes. A global beamsplitter is specified by two natural numbers, $k \neq l$, between 1 and n. The global beamsplitter then acts on all the modes with indices $\{m\}$ such that their k-th bit is equal to 0, by applying local beamsplitters between modes with indices m, m' that only differ in their l-th bit.

Our decision problem is then phrased as follows.

Problem 7.1
Consider a bit-structured interferometer (see Definition 7.1) acting on 2^n modes with $L \in \mathcal{O}(\mathrm{poly}(n))$, and an input state such that the first mode is displaced in position by a real constant x while the state of the remaining modes is the vacuum. Then, decide whether the expectation value of the position on the first mode at the output of the interferometer is

$$1.\, \langle \hat{q}_1 \rangle > \frac{2}{3}x\,, \qquad \text{or} \qquad 2.\, \langle \hat{q}_1 \rangle < \frac{1}{3}x\,,$$

given the promise that either one or the other is true.

Problem 7.1 is illustrated in Figure 7.3, where we simulate an interferometer with ~ 8 billion modes (i.e., a 33-qubit circuit). There, we keep track of $\langle \hat{q}_1 \rangle / x$ (which corresponds to the overlap with the $|0\rangle^{\otimes n}$ state) as the state evolves through the beamsplitters.

7.3.2. Complexity of the problem

Our main result is the next theorem.

Theorem 7.1
Problem 7.1 is BQP-complete.

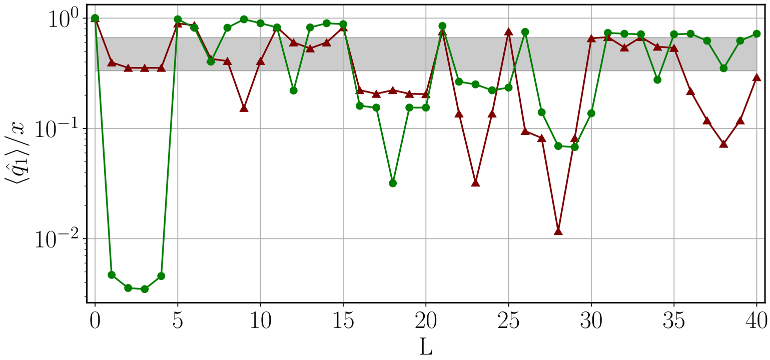


Figure 7.3.: **Simulation of a bit-structured interferometer on ~ 8 billion modes**. We illustrate Problem 7.1 by tracking two non-trivial evolutions of $\langle \hat{q}_1 \rangle / x$ along a large bit-structured interferometer, the green (red) plot corresponding to a YES (NO) instance. The gray region corresponds to $\frac{1}{3} < \langle \hat{q}_1 \rangle / x < \frac{2}{3}$. The simulations were performed with the open-source library `Qibo` [195, 196].

The proof of Theorem 7.1 can be found in the SI, but we give here a summary of the key points. First, we show that Problem 7.1 is contained in BQP. To do this, we simply use the mapping between beamsplitters and qubit gates explained previously to prove that a bit-structured interferometer with $\mathcal{O}(\text{poly}(n))$ layers always results in a polynomial-size quantum circuit, which is a necessary condition to be contained in BQP.

Finally, we need to prove that Problem 7.1 is BQP-hard. That is, if we can solve it, we can also solve any other problem in BQP with an additional overhead that is polynomial in n. This is done by first showing that each global beamsplitter gives rise to a controlled-R_y gate (acting when the control qubit is in the $|0\rangle$ state). Then, we use the result from [197] which states that controlled-R_y rotations constitute a universal gate set, in the sense that any quantum computation can be performed with these gates [198, 199].

7.4. Non-energy-preserving systems

7.4.1. Quantum algorithms for dissipative exponentially large differential equations

The main result of the previous section is to propose a new BQP-complete decision problem based on exponentially large interferometers. Other energy-preserving exponentially large systems [42, 200] yield BQP-complete decision problems. A natural question is whether non-energy-conserving systems can also be efficiently simulated. The answer is partly found in the literature of quantum algorithms to solve differential equations. Most results [174, 201–203] are based on the Harrow-Hassidim-Lloyd (HHL) [171] subroutine. Caveats for the HHL algorithm to provide an actual advantage have been well studied [173], in particular, the algorithm's scaling with the condition number of the matrix to invert.

In [174], a lower bound for the complexity of any quantum algorithm returning the solution to a differential equation $\partial_t x = Ax$ as an amplitude encoded state $|x\rangle = \frac{1}{\|x\|_2}\sum_k x_k |k\rangle$ was provided. For some dissipative systems this complexity bound scales exponentially with the integration time. More precisely, noting that any stable system must have eigenvalues with non-positive real parts (otherwise it would diverge exponentially), the difference between the maximum and the minimum real part of eigenvalues $\{i\omega_k - \eta_k\}$ of A is defined as $\delta := |\mathrm{max}_k \eta_k - \mathrm{min}_k \eta_k|$. In [174], the worst case complexity to evolve the system for a time t is shown to be $\Omega(e^{\delta t})$, with a proof based on the distinguishability of non-orthogonal quantum states. This lower bound can also be understood in terms of the condition number of e^{At} [203] which is ultimately the linear transformation that is applied to the initial state. Indeed, considering that all eigenstates of A are orthogonal for simplicity, we can write $A = \sum (i\omega_k - \eta_k) |k\rangle\langle k|$, and find its condition number to be

$$\begin{aligned}\kappa(e^{At}) = \frac{|\lambda_{\mathrm{max}}|}{|\lambda_{\mathrm{min}}|} &= \frac{\mathrm{max}_k(|e^{(i\omega_k - \eta_k)t}|)}{\mathrm{min}_k(|e^{(i\omega_k - \eta_k)t}|)} \\ &= e^{t(\mathrm{min}_k(\eta_k) - \mathrm{max}_k(\eta_k))} = e^{\delta t} \end{aligned} \tag{7.2}$$

This means that the dissipation of energy yields a surge in the complexity of any quantum algorithm to be exponential in the time of integration. Note that for constant time, the complexity of the algorithm may remain polynomial, which is in line with existing literature about dissipative systems [172]. In contrast, time scaling at least linearly with the number of qubits, that is logarithmically with the ODE dimension, may increase the complexity to be exponential. Such a complexity increase in the presence of dissipative terms is also found for classical algorithms solving ordinary differential equations. Although there is no strict mathematical definition, some ODEs are called stiff, when in practice their numerical resolution suffers from instability. In particular, in that context for linear ODEs, the stiffness ratio is defined as $\min_k(\eta_k)/\max_k(\eta_k)$. In summary, previous work has shown that energy dissipation in ODEs leads to an increase in the complexity of solving them with a quantum computer for integration times scaling logarithmically with the ODE dimension. In this section, we quantify such an increase by proving hardness results. We define the following problem.

Problem 7.2
Given a system of ODEs, whose dynamics are characterized by a matrix $A \in \mathbb{R}(2^n, 2^n)$*, such that its spectral norm* $\|A\| \leq R \in O(poly(n))$*, a time* t*, an input state* x_0 *output the state* $x(t) = \frac{e^{At}x_0}{\|e^{At}x_0\|_2}$.

The first step in analyzing the complexity of this problem is to realize that the imaginary-time evolution of a real Hamiltonian can be reduced to it. Indeed given a real Hamiltonian H, define $A = -H$ and $t = \beta$ and one has $e^{-\beta H} = e^{At}$. Notice that the converse is not true as A may not be symmetric, i.e. not a Hamiltonian.

7.4.2. The power of imaginary-time evolution

Imaginary-time evolution is a powerful tool. Aside heuristic approaches based on variational principles [81], two types of algorithms to perform imaginary-time evolution with theoretical guarantees exist. The approach in [194] is probabilistic, it is a sequence of steps each one involving post-selection. Another approach is Quantum Imaginary Time evolution (QITE) [191], which consists of approximating small steps of imaginary-time evolution with a time-dependent unitary, whose parameters are derived by performing tomography on the state. The complexity of both quantum algorithms is super-polynomial in general. We define the exact imaginary-time evolution problem as follows.

Problem 7.3
Given a Hamiltonian H *with spectral norm* $\|H\| \in poly(n)$*, a time* β*, an input state* $|\psi_0\rangle$ *prepare the state* $|\psi_t\rangle = \frac{e^{-\beta H}|\psi_0\rangle}{\|e^{-\beta H}|\psi_0\rangle\|_2}$.

In [34] PostBQP is proven to be equal to PP. Informally, PostBQP is the set of decision problems that one can solve with BQP equipped with post-selection, problems that can be solved with a probabilistic algorithm that gives the correct answer with a probability of failure arbitrarily close to 1/2. In fact it follows from Proposition 5 in [34] that imaginary-time evolution combined with real-time evolution has a computational power equivalent to that of PostBQP, as it is allows for all invertible transformations, not only just unitary. We give an intuition on why imaginary-time evolution gives access to post-selection of events, even those with exponentially small probability. Consider that for $\varepsilon = e^{-q(n)}$ with q a polynomial, one is given the following state

$$|\phi\rangle = \sqrt{\varepsilon}\,|0\rangle\,|\phi_0\rangle + \sqrt{1-\varepsilon}\,|1\rangle\,|\phi_1\rangle\ . \tag{7.3}$$

The task is to post-select on the first qubit being $|0\rangle$. This has an exponentially small probability ε of occurring. We use access to imaginary-time evolution (see Problem 7.3) with input state $|\phi\rangle$, Hamiltonian $H = -Z \otimes I$ and time β, which yields the state

$$|\psi\rangle = \frac{e^{+\beta}\sqrt{\varepsilon}\,|0\rangle\,|\phi_0\rangle + e^{-\beta}\sqrt{1-\varepsilon}\,|1\rangle\,|\phi_1\rangle}{\sqrt{e^{+2\beta}\varepsilon + e^{-2\beta}(1-\varepsilon)}}. \tag{7.4}$$

Finally choosing $\beta = q(n)/2$ the probability p_0 of measuring the first qubit in $|0\rangle$ is very close to 1, as follows.

$$p_0 = \frac{e^{+2\beta}\varepsilon}{e^{+2\beta}\varepsilon + e^{-2\beta}(1-\varepsilon)} > \frac{1}{1+e^{-q(n)}}\ . \tag{7.5}$$

Therefore for time $\beta \in \Omega(\mathrm{poly}(n))$ imaginary-time evolution combined with real-time evolution is PostBQP-complete. Could one find a lower complexity class for lower integration time say, $\beta \in \Theta(\log(n))$?

It is well known that imaginary-time evolution is used to solve ground-state problems, including QMA-complete problems. For completeness we provide the constructive reduction of a QMA-complete problem to the imaginary-time evolution, in Section 7.6.5. At a high level we show that given an instance of the ground-energy problem for a 2-local real Hamiltonian [204] H over n qubits, one can use access to a solver of the imaginary-time evolution problem (Problem 7.3) with $|\psi_0\rangle$ taken at random (e.g. from a 2-design), H and $\beta \in \mathrm{poly}(n)$ to solve the ground-energy problem in polynomial time. The proof yields a lower bound on the time of integration, sufficient to guarantee that the ground energy problem can be resolved. This bound is tight, there exist a Hamiltonian for which this lower bound is also necessary. This lower bound on the integration time is inversely proportional to the promise gap, which is itself at most polynomially small. The proof in Section 7.6.5 is super-seeded by the post-selection argument because QMA$\subseteq$PostBQP. In addition, because of this lower bound there is no scaling of β that would make the problem QMA-hard but not PostBQP-complete.

7.4.3. PostBQP-completeness of exponentially large non-energy-conserving systems

Combining the previous, we have that post-selection can be reduced to imaginary-time evolution for polynomial integration time β, which can in turn be reduced to non-energy-conserving exponentially large differential equations. With this we can conclude that the latter is PostBQP-hard for times poly-logarithmic in the system size (i.e, polynomial in the number of qubits n). We have also identified that squeezing gates in our framework correspond to imaginary time evolution of the $Z \otimes |m\rangle\langle m|$ Hamiltonian in Section 7.6.2. This is precisely the Hamitonian we use in Section 7.4.2 to illustrate how one may use imaginary-time evolution to perform post-selection. That means that adding squeezing gates to the exponentially large interferometer boosts its complexity from BQP-complete to PostBQP-complete, a dramatic surge in computational power (we provide more details in Section 7.6.6).

7.5. Outlook

Our work contributed to the body of knowledge of schemes leading to BQP-complete tasks. In particular, along with Ref. [42], we show examples of how linear and energy-preserving evolutions of exponentially many simple physical systems can be efficiently simulated on a quantum computer. A key unique feature of our framework is that we translate a sequence of gates from one physical system to another (product of exponentials) rather than performing real-time evolution of a Hamiltonian (exponential of a sum).

This main part of our work opens up several interesting research directions. For instance, we have shown how to simulate the evolution of the first and second moments of bosonic states. This makes the simulation complete only for Gaussian states. Hence, we envision two possible paths to extend our proposed framework to better approximate the simulation of non-Gaussian states. First, we could find ways to simulate the evolution of higher moments. To what extent this would improve the quality of a simulation for non-Gaussian states, such as Fock states, remains open.

Second, because bosonic states can be written as a continuous sum over coherent states, approximating them on a grid of coherent states may also be a viable strategy to simulate non-Gaussian states.

Additionally, we also show that for time poly-logarithmic in the system size a set of non-energy-conserving systems yield PostBQP-completeness. This is compatible with existing literature which devises quantum algorithms that are efficient for constant times only [172]. Moreover, we prove that adding squeezing gates boosts the computational power of interferometers from BQP-complete to PostBQP-complete.

We hope that this chapter sharpens the community's understanding of the power of quantum computers to simulate large dynamical linear systems. The next frontier is non-linear systems, which are particularly relevant because they include computational fluid dynamics. Can we classify some non-linear classical dynamical problems with respect to quantum complexity classes such as BQP, QMA or PostBQP?

7.6. Appendix

7.6.1. Framework

Time evolution of quadrature operators in phase space

Let us consider a system of bosons with M modes, and let us assume $\hbar = 1$. The state space of such a system is $\mathcal{H} = \bigotimes_{m=1}^{M} \mathcal{F}_m$, where $\mathcal{F}_m$ is the Fock space associated to the m-th mode. That is, each $\mathcal{F}_m$ is spanned by the infinitely-many basis vectors $\{|k\rangle_m\}_{k\in\mathbb{N}}$ indicating the occupancy number of the mode. For example, if we fix the number of bosons to 3 and let the number of modes be $M = 4$, the basis state $|0,1,2,0\rangle \equiv |0\rangle \otimes |1\rangle \otimes |2\rangle \otimes |0\rangle$ corresponds to the state with one boson occupying the second mode and two bosons occupying the third mode. These basis states are eigenstates of the particle number operator, since they have a fixed number of particles, and are known as Fock states. The (non-Hermitian) creation $\hat{a}_m^\dagger$ and annihilation $\hat{a}_m$ operators are defined by their action on Fock states as follows,

$$\begin{aligned} \hat{a}_m^\dagger |N_0,\ldots,N_m,\ldots,N_M\rangle &= \sqrt{N_m+1}\; |N_0,\ldots,N_m+1,\ldots,N_M\rangle \ , \\ \hat{a}_m |N_0,\ldots,N_m,\ldots,N_M\rangle &= \sqrt{N_m}\; |N_0,\ldots,N_m-1,\ldots,N_M\rangle \ . \end{aligned} \tag{7.6}$$

They satisfy the commutation relations

$$\begin{aligned}[\hat{a}^\dagger_m, \hat{a}^\dagger_{m'}] &= [\hat{a}_m, \hat{a}_{m'}] = 0\,, \\ [\hat{a}_m, \hat{a}^\dagger_{m'}] &= \delta_{m,m'}\,.\end{aligned} \tag{7.7}$$

The (Hermitian) particle number operator is given by $\hat{a}^\dagger_m \hat{a}_m$. The (Hermitian) position $\hat{q}_m$ and momentum $\hat{p}_m$ operators can be defined from the creation and annihilation operators as

$$\hat{q}_m = \frac{\hat{a}_m + \hat{a}^\dagger_m}{\sqrt{2}}\,, \qquad \hat{p}_m = i\,\frac{\hat{a}^\dagger_m - \hat{a}_m}{\sqrt{2}}\,. \tag{7.8}$$

Which in turn conversely implies that

$$\hat{a}_m = \frac{\hat{q}_m + i\hat{p}_m}{\sqrt{2}}\,, \qquad \hat{a}^\dagger_m = \frac{\hat{q}_m - i\hat{p}_m}{\sqrt{2}}\,. \tag{7.9}$$

The corresponding commutation relations for position and momentum are

$$\begin{aligned}[\hat{q}_m, \hat{q}_{m'}] &= [\hat{p}_m, \hat{p}_{m'}] = 0\,, \\ [\hat{q}_m, \hat{p}_{m'}] &= i\delta_{m,m'}\,.\end{aligned} \tag{7.10}$$

Let us collect the positions and momenta in a vector

$$\hat{\vec{z}} = (\hat{q}_1, \ldots, \hat{q}_M, \hat{p}_1, \ldots, \hat{p}_M)^T$$

. This allows us to write the commutation relations in Equation (7.10) as

$$\left[\hat{\vec{z}}, \hat{\vec{z}}^T\right] = i\Omega\,, \tag{7.11}$$

where the Ω matrix is

$$\Omega = \begin{pmatrix} 0 & I_M \\ -I_M & 0 \end{pmatrix} = iY \otimes I_M\,. \tag{7.12}$$

Here, Y is the usual 2×2 Pauli matrix and I_M the $M \times M$ identity matrix. It will be convenient for us to explicitly write down the matrix entries of Ω, which are

$$\Omega_{\gamma,\gamma'} = \delta_{\gamma,\gamma'-M} - \delta_{\gamma,\gamma'+M}\,. \tag{7.13}$$

Let us assume that our quantum bosonic system is governed by a quadratic time-independent Hamiltonian of the form

$$\hat{H} = \frac{1}{2}\hat{\vec{z}}^T K \hat{\vec{z}}, \tag{7.14}$$

where K is a real $2M \times 2M$ symmetric matrix. In the Heisenberg picture, the equation of motion of an observable $\hat{O}$ is

$$\frac{\partial \hat{O}}{\partial t} = i[\hat{H}, \hat{O}]\,. \tag{7.15}$$

Therefore, we find

$$\begin{aligned}
\frac{\partial \hat{\vec{z}}_\gamma}{\partial t} &= i[\hat{H}, \hat{\vec{z}}_\gamma] = \frac{i}{2}\left[\sum_{\alpha\beta} \hat{\vec{z}}_\alpha K_{\alpha\beta} \hat{\vec{z}}_\beta, \hat{\vec{z}}_\gamma\right] \\
&= \frac{i}{2}\sum_{\alpha\beta} K_{\alpha\beta}\left[\hat{\vec{z}}_\alpha \hat{\vec{z}}_\beta, \hat{\vec{z}}_\gamma\right] = \frac{i}{2}\sum_{\alpha\beta} K_{\alpha\beta}\left(\hat{\vec{z}}_\alpha\left[\hat{\vec{z}}_\beta, \hat{\vec{z}}_\gamma\right] + \left[\hat{\vec{z}}_\alpha, \hat{\vec{z}}_\gamma\right]\hat{\vec{z}}_\beta\right) \\
&= \frac{i}{2}\sum_{\alpha\beta} K_{\alpha\beta}\left(\hat{\vec{z}}_\alpha i(\delta_{\beta,\gamma-M} - \delta_{\beta,\gamma+M}) + i(\delta_{\alpha,\gamma-M} - \delta_{\alpha,\gamma+M})\hat{\vec{z}}_\beta\right) \\
&= \frac{1}{2}\sum_{\alpha\beta} K_{\alpha\beta}\left(-\hat{\vec{z}}_\alpha \Omega_{\beta\gamma} - \Omega_{\alpha\gamma}\hat{\vec{z}}_\beta\right) = \frac{1}{2}\sum_{\alpha\beta}\left(\Omega_{\gamma\beta}K_{\beta\alpha}\hat{\vec{z}}_\alpha + \Omega_{\gamma\alpha}K_{\alpha\beta}\hat{\vec{z}}_\beta\right) \\
&= \left(\Omega K \hat{\vec{z}}\right)_\gamma\,,
\end{aligned} \tag{7.16}$$

where we used $[AB, C] = A[B, C] + [A, C]B$, and the fact that K is symmetric and Ω anti-symmetric. Hence, we arrive at

$$\frac{\partial \hat{\vec{z}}}{\partial t} = \Omega K \hat{\vec{z}}. \tag{7.17}$$

The solution to this differential equation is given by

$$\hat{\vec{z}}(t) = e^{t\Omega K}\hat{\vec{z}}(0)\,. \tag{7.18}$$

The solution of Equation (7.17) must preserve the commutation relations of Equation (7.10) in order to leave the kinematics invariant. Let us call $Q(t) = e^{t\Omega K}$ the propagator that takes the vector $\hat{\vec{z}}(0)$ at time 0 to the vector $\hat{\vec{z}}(t)$ at time t. We impose $\left[\hat{\vec{z}}(t), \hat{\vec{z}}(t)^T\right] = [\hat{\vec{z}}(0), \hat{\vec{z}}(0)^T]$, which leads to

$$\left[Q(t)\hat{\vec{z}}(0), (Q(t)\hat{\vec{z}}(0))^T\right]_{\alpha\beta} = \left[\left(\sum_{\gamma} Q(t)_{\alpha\gamma}\hat{\vec{z}}_{\gamma}\right), \left(\sum_{\gamma'} Q(t)_{\beta\gamma'}\hat{\vec{z}}_{\gamma'}\right)\right] \tag{7.19}$$

$$= \sum_{\gamma,\gamma'} Q(t)_{\alpha\gamma}Q(t)_{\beta\gamma'}\left[\hat{\vec{z}}_{\gamma}, \hat{\vec{z}}_{\gamma'}\right] \tag{7.20}$$

$$= i\sum_{\gamma,\gamma'} Q(t)_{\alpha\gamma}Q(t)_{\beta\gamma'}(\delta_{\gamma,\gamma'-M} - \delta_{\gamma,\gamma'+M}) \tag{7.21}$$

$$= i\sum_{\gamma,\gamma'} Q(t)_{\alpha\gamma}\Omega_{\gamma,\gamma'}Q(t)^T_{\gamma'\beta} \tag{7.22}$$

$$= i\left(Q(t)\,\Omega\,Q^T(t)\right)_{\alpha\beta}\,. \tag{7.23}$$

It is clear then that $Q(t)\,\Omega\,Q^T(t) = \Omega$ if we are to maintain the commutation relations between positions and momenta. This is precisely the defining condition of symplectic matrices. Hence, the time evolution of $\hat{\vec{z}}$ in phase space is given by a $2M \times 2M$ real symplectic matrix belonging to the group $\mathbb{SP}(M,\mathbb{R})$.

Let us now address the question of how the expectation value $\langle\hat{\vec{z}}\rangle = (\langle\hat{q}_1\rangle, \ldots, \langle\hat{q}_M\rangle, \langle\hat{p}_1\rangle, \ldots, \langle\hat{p}_M\rangle)^T$ of $\hat{\vec{z}}$ evolves in time, given an initial quantum state ρ_0. We find that

$$\frac{\partial\langle\hat{\vec{z}}_{\gamma}\rangle}{\partial t} = \frac{\partial\ \mathrm{Tr}\left[\hat{\vec{z}}_{\gamma}\rho_0\right]}{\partial t} = \mathrm{Tr}\left[\frac{\partial\hat{\vec{z}}_{\gamma}}{\partial t}\rho_0\right] = \mathrm{Tr}\left[(\Omega K\hat{\vec{z}})_{\gamma}\rho_0\right] =$$
$$= \mathrm{Tr}\left[\left(\sum_{\alpha\beta}\Omega_{\gamma\alpha}K_{\alpha\beta}\hat{\vec{z}}_{\beta}\right)\rho_0\right] = \sum_{\alpha\beta}\Omega_{\gamma\alpha}K_{\alpha\beta}\,\mathrm{Tr}\left[\hat{\vec{z}}_{\beta}\rho_0\right] = (\Omega K\langle\hat{\vec{z}}\rangle)_{\gamma}\,, \tag{7.24}$$

which implies

$$\frac{\partial\langle\hat{\vec{z}}\rangle}{\partial t} = \Omega K\langle\hat{\vec{z}}\rangle\,, \tag{7.25}$$

and

$$\langle\hat{\vec{z}}\rangle(t) = e^{t\Omega K}\langle\hat{\vec{z}}\rangle(0)\,. \tag{7.26}$$

We can also collect the expectation value of products of quadrature operators over ρ_0 in the $2M \times 2M$ positive-definite covariance matrix $\vec{\sigma}$

whose entries are given by

$$\vec{\sigma}_{\alpha\beta} = \frac{1}{2}\langle \hat{\vec{z}}_\alpha \hat{\vec{z}}_\beta + \hat{\vec{z}}_\beta \hat{\vec{z}}_\alpha \rangle - \langle \hat{\vec{z}}_\alpha \rangle \langle \hat{\vec{z}}_\beta \rangle \,. \tag{7.27}$$

Let us derive the corresponding equation of motion. We start by considering

$$\begin{aligned}
\frac{\partial(\hat{\vec{z}}_\gamma \hat{\vec{z}}_\delta)}{\partial t} &= i[\hat{H}, \hat{\vec{z}}_\gamma \hat{\vec{z}}_\delta] = \frac{i}{2}\Big[\sum_{\alpha\beta} \hat{\vec{z}}_\alpha K_{\alpha\beta} \hat{\vec{z}}_\beta, \hat{\vec{z}}_\gamma \hat{\vec{z}}_\delta\Big] = \frac{i}{2}\sum_{\alpha\beta} K_{\alpha\beta}\left[\hat{\vec{z}}_\alpha \hat{\vec{z}}_\beta, \hat{\vec{z}}_\gamma \hat{\vec{z}}_\delta\right] \\
&= \frac{i}{2}\sum_{\alpha\beta} K_{\alpha\beta}\left(\hat{\vec{z}}_\alpha \left[\hat{\vec{z}}_\beta, \hat{\vec{z}}_\gamma\right]\hat{\vec{z}}_\delta + \left[\hat{\vec{z}}_\alpha, \hat{\vec{z}}_\gamma\right]\hat{\vec{z}}_\beta \hat{\vec{z}}_\delta\right. \qquad (7.28) \\
&\left. + \hat{\vec{z}}_\gamma \hat{\vec{z}}_\alpha \left[\hat{\vec{z}}_\beta, \hat{\vec{z}}_\delta\right] + \hat{\vec{z}}_\gamma \left[\hat{\vec{z}}_\alpha, \hat{\vec{z}}_\delta\right]\hat{\vec{z}}_\beta\right) \\
&= -\frac{1}{2}\sum_{\alpha\beta} K_{\alpha\beta}\left(\hat{\vec{z}}_\alpha(\delta_{\beta,\gamma-M} - \delta_{\beta,\gamma+M})\hat{\vec{z}}_\delta + (\delta_{\alpha,\gamma-M} - \delta_{\alpha,\gamma+M})\hat{\vec{z}}_\beta \hat{\vec{z}}_\delta\right. \\
&\left. + \hat{\vec{z}}_\gamma \hat{\vec{z}}_\alpha(\delta_{\beta,\delta-M} - \delta_{\beta,\delta+M}) + \hat{\vec{z}}_\gamma(\delta_{\alpha,\delta-M} - \delta_{\alpha,\delta+M})\hat{\vec{z}}_\beta\right) \\
&= -\frac{1}{2}\sum_{\alpha\beta} K_{\alpha\beta}\left(\hat{\vec{z}}_\alpha \Omega_{\beta\gamma}\hat{\vec{z}}_\delta + \Omega_{\alpha\gamma}\hat{\vec{z}}_\beta \hat{\vec{z}}_\delta + \hat{\vec{z}}_\gamma \hat{\vec{z}}_\alpha \Omega_{\beta\delta} + \hat{\vec{z}}_\gamma \Omega_{\alpha\delta}\hat{\vec{z}}_\beta\right) \\
&= -(\hat{\vec{z}}^T K\Omega)_\gamma \hat{\vec{z}}_\delta - \hat{\vec{z}}_\gamma(\hat{\vec{z}}^T K\Omega)_\delta = (\Omega K\hat{\vec{z}})_\gamma \hat{\vec{z}}_\delta + \hat{\vec{z}}_\gamma(\Omega K\hat{\vec{z}})_\delta \,,
\end{aligned}$$

where we used that $[AB, CD] = A[B,C]D + [A,C]BD + CA[B,D] + C[A,D]B$, and the fact that K (Ω) is symmetric (anti-symmetric). In matrix form, we get

$$\frac{\partial(\hat{\vec{z}}\hat{\vec{z}}^T)}{\partial t} = \Omega K \hat{\vec{z}}\hat{\vec{z}}^T - \hat{\vec{z}}\hat{\vec{z}}^T K\Omega \,. \tag{7.29}$$

Let us now look at the evolution of the expectation value of two-point correlators. Analogously to Equation (7.24), we have

$$\begin{aligned}
\frac{\partial\langle \hat{\vec{z}}_\gamma \hat{\vec{z}}_\delta\rangle}{\partial t} &= \frac{\partial \operatorname{Tr}\left[\hat{\vec{z}}_\gamma \hat{\vec{z}}_\delta \rho\right]}{\partial t} = \operatorname{Tr}\left[\frac{\partial \hat{\vec{z}}_\gamma}{\partial t}\hat{\vec{z}}_\delta \rho + \hat{\vec{z}}_\gamma \frac{\partial \hat{\vec{z}}_\delta}{\partial t}\rho\right] \\
&= \operatorname{Tr}\left[(\Omega K \hat{\vec{z}})_\gamma \hat{\vec{z}}_\delta \rho + \hat{\vec{z}}_\gamma (\Omega K \hat{\vec{z}})_\delta \rho\right] \\
&= \operatorname{Tr}\left[\left(\sum_{\alpha\beta} \Omega_{\gamma\alpha} K_{\alpha\beta} \hat{\vec{z}}_\beta\right) \hat{\vec{z}}_\delta \rho + \hat{\vec{z}}_\gamma \left(\sum_{\alpha\beta} \Omega_{\delta\alpha} K_{\alpha\beta} \hat{\vec{z}}_\beta\right) \rho\right] \\
&= \sum_{\alpha\beta} \Omega_{\gamma\alpha} K_{\alpha\beta} \operatorname{Tr}\left[\hat{\vec{z}}_\beta \hat{\vec{z}}_\delta \rho\right] + \sum_{\alpha\beta} \Omega_{\delta\alpha} K_{\alpha\beta} \operatorname{Tr}\left[\hat{\vec{z}}_\gamma \hat{\vec{z}}_\beta \rho\right] \\
&= (\Omega K \langle \hat{\vec{z}} \hat{\vec{z}}^T \rangle)_{\gamma\delta} - (\langle \hat{\vec{z}} \hat{\vec{z}}^T \rangle K \Omega)_{\gamma\delta}\,, \qquad (7.30)
\end{aligned}$$

or, in matrix form,

$$\frac{\partial(\langle \hat{\vec{z}} \hat{\vec{z}}^T \rangle)}{\partial t} = \Omega K \langle \hat{\vec{z}} \hat{\vec{z}}^T \rangle - \langle \hat{\vec{z}} \hat{\vec{z}}^T \rangle K \Omega\,. \qquad (7.31)$$

Therefore, we arrive at

$$\frac{\partial \vec{\sigma}}{\partial t} = \Omega K \vec{\sigma} - \vec{\sigma} K \Omega\,. \qquad (7.32)$$

If K represents a particle-preserving Hamiltonian, then $[\Omega, K] = 0$ according to Proposition 7.2, so we can write

$$\frac{\partial \vec{\sigma}}{\partial t} = [\Omega K, \vec{\sigma}]\ . \qquad (7.33)$$

The solution to this equation is

$$\vec{\sigma} = e^{\Omega K t}\, \vec{\sigma}(0)\, e^{-\Omega K t}\,. \qquad (7.34)$$

If instead K represents a non-particle-preserving Hamiltonian such that $\{\Omega, K\} = 0$, we can write

$$\frac{\partial \vec{\sigma}}{\partial t} = \{\Omega K, \vec{\sigma}\}\ , \qquad (7.35)$$

whose solution is

$$\vec{\sigma}(t) = e^{\Omega K t}\, \vec{\sigma}(0)\, e^{\Omega K t}\,. \qquad (7.36)$$

Pauli basis for the symplectic algebra

Here we present a useful Supplemental Proposition that provides a Pauli basis for the symplectic algebra $\mathfrak{sp}(M,\mathbb{R})$.

Proposition 7.1
An orthogonal basis for the standard representation of the $\mathfrak{sp}(M,\mathbb{R})$ algebra, where $M=2^n$, is given by the set

$$B_{\mathfrak{sp}(M,\mathbb{R})} \equiv i\{Y\otimes P_s\} \cup i\{I\otimes P_a\} \cup \{X\otimes P_s\} \cup \{Z\otimes P_s\}\,, \quad (7.37)$$

where P_s and P_a belong to the sets of arbitrary symmetric and anti-symmetric Pauli strings on n qubits, respectively, and I,X,Y,Z are the usual 2×2 Pauli matrices.

Proof. We first recall that any $2M\times 2M$ matrix A satisfying $A^T\Omega=-\Omega A$ belongs to $\mathfrak{sp}(M,\mathbb{R})\subset \mathrm{End}(\mathbb{R}^{2M})$. Clearly, the (phased) Pauli operators on $n+1$ qubits in Equation (7.37) constitute $2^{n+1}\times 2^{n+1}=2M\times 2M$ matrices, with $M=2^n$. Also, note they are all real-valued (although not all anti-Hermitian), which follows from the fact that a Pauli string is real (purely imaginary) when it contains an even (odd) number of Y's, i.e. when it is symmetric (anti-symmetric). Thus, $B_{\mathfrak{sp}(M,\mathbb{R})}\subset \mathrm{End}(\mathbb{R}^{2M})$. We know from Proposition 1 in Ref. [190] that they satisfy the symplectic property, implying $B_{\mathfrak{sp}(M,\mathbb{R})}\subset \mathfrak{sp}(M,\mathbb{R})$. Given that they are Hilbert-Schmidt orthogonal, and that $|B_{\mathfrak{sp}(M,\mathbb{R})}|=M(2M+1)=\dim(\mathfrak{sp}(M,\mathbb{R}))$, to prove they constitute an orthogonal basis for $\mathfrak{sp}(M,\mathbb{R})$ it simply remains to show that they are closed under commutation.

To do so, we first recall that the commutator of two anti-symmetric or symmetric matrices is anti-symmetric, whereas the commutator of a symmetric matrix and an anti-symmetric one is symmetric. We start with the (non-zero) commutator $X\otimes P_s$ and $X\otimes P_s'$, which gives an operator of the following form

$$[X\otimes P_s, X\otimes P_s'] \propto \pm i I\otimes P_a\,. \quad (7.38)$$

The $\pm i$ factor follows from the fact that the Pauli strings P_s and P_s' differ at an odd number of sites. The same is true if we replace X by Z on the first qubit,

$$[Z\otimes P_s, Z\otimes P_s'] \propto \pm i I\otimes P_a\,. \quad (7.39)$$

We continue by computing the (non-zero) commutator of two operators of the form $X\otimes P_s$ and $Z\otimes P_s'$,

$$[X\otimes P_s, Z\otimes P_s'] \propto \pm i Y\otimes P_s''\,. \quad (7.40)$$

Again, the $\pm i$ factor follows from the fact that $X \otimes P_s$ and $Z \otimes P_s'$ differ at an odd number of sites. Next, let us look at the non-zero commutator of operators $X \otimes P_s$ or $Z \otimes P_s$ with $iI \otimes P_a$,

$$[X \otimes P_s, iI \otimes P_a] \propto \pm X \otimes P_s' \quad \text{or} \quad [Z \otimes P_s, iI \otimes P_a] \propto \pm Z \otimes P_s' \,. \tag{7.41}$$

Here, the i factor arising from commuting the Pauli strings cancels out with the i in $iI \otimes P_a$. Similarly, the non-zero commutator of $X \otimes P_s$ or $Z \otimes P_s$ with $iY \otimes P_s'$ is as follows

$$[X \otimes P_s, iY \otimes P_s'] \propto \pm Z \otimes P_s'' \quad \text{or} \quad [Z \otimes P_s, iY \otimes P_s'] \propto \pm X \otimes P_s'' \,. \tag{7.42}$$

Furthermore, commuting $iY \otimes P_s$ and $iY \otimes P_s'$, or $iI \otimes P_a$ with $iI \otimes P_a'$ leads to either zero or an operator of the form

$$[iY \otimes P_s, iY \otimes P_s'] \propto \pm iI \otimes P_a \quad \text{or} \quad [iI \otimes P_a, iI \otimes P_a'] \propto \pm iI \otimes P_a'' \,. \tag{7.43}$$

Finally, the non-zero commutator of $iY \otimes P_s$ and $iI \otimes P_a$ gives $\pm iY \otimes P_s$,

$$[iY \otimes P_s, iI \otimes P_a] \propto \pm iY \otimes P_s \,. \tag{7.44}$$

Therefore, we conclude that the set $B_{\mathfrak{sp}(M,\mathbb{R})} \subset \mathfrak{sp}(M,\mathbb{R})$ of mutually orthogonal operators is closed under commutation and satisfies

$$\dim(B_{\mathfrak{sp}(M,\mathbb{R})}) = M(2M+1) = \dim(\mathfrak{sp}(M,\mathbb{R})) \,.$$

Thus, it constitutes an orthogonal basis for the standard representation of $\mathfrak{sp}(M,\mathbb{R})$. ☐

Particle-preserving gates

Next, we show that particle-preserving Gaussian bosonic (GB) gates lead to unitary evolutions at the qubit level.

Proposition 7.2
When a gate generator is of the form

$$\hat{H} = \sum_{m,m'=1}^{M} h_{mm'} \, \hat{a}_m^\dagger \hat{a}_{m'} + \frac{\mathrm{Tr}[h]}{2} I_{2M} \,, \tag{7.45}$$

where h is a Hermitian matrix, then $[\Omega, K] = 0$, and the propagator $e^{t\Omega K}$

is the real time evolution under the effective Hamiltonian $-i\Omega K$.

Proof. Expressed in terms of position and momentum operators, we find

$$\begin{aligned}\hat{H} &= \frac{1}{2}\sum_{m,m'=1}^{2^n} h_{mm'}\,(q_m - ip_m)(q_{m'} + ip_{m'}) + \frac{\mathrm{Tr}[h]}{2} I_{2M} \\ &= \frac{1}{2}\sum_{m,m'=1}^{2^n} h_{mm'}\,(q_m q_{m'} + p_m p_{m'} + iq_m p_{m'} - ip_m q_{m'}) + \frac{\mathrm{Tr}[h]}{2} I_{2M} \\ &= \frac{1}{2}\sum_{m=1}^{2^n} h_{mm}\,(q_m^2 + p_m^2) + \sum_{\substack{m,m'=1\\ m'>m}}^{2^n} \mathrm{Re}[h_{mm'}]\,(q_m q_{m'} + p_m p_{m'}) \\ &+ \sum_{\substack{m,m'=1\\ m'>m}}^{2^n} \mathrm{Im}[h_{mm'}]\,(q_m p_{m'} - p_m q_{m'})\,, \end{aligned} \tag{7.46}$$

where we used the commutation relations from Equation (7.10). In other words, particle-preserving gate generators are such that the K matrix is a real linear combination of Paulis of the form $I \otimes P_s$ (corresponding to the first two sums in Equation (7.46)) and/or $Y \otimes P_a$ (corresponding to the last sum in Equation (7.46)). This automatically implies that $[\Omega, K] = 0$. Finally, either ΩK is a real combination of Paulis of the form $(iY \otimes I_M)(I \otimes P_s) = iY \otimes P_s$ or $(iY \otimes I_M)(Y \otimes P_a) = iI \otimes P_a$ (both of which are in the symplectic algebra according to Proposition 7.1). That is, ΩK is anti-Hermitian and the symplectic propagator $e^{t\Omega K}$ is unitary. Hence, these types of gate generators result in unitary dynamics in phase space. ☐

Non-particle-preserving gates

We here show that a family of non-particle-preserving GB gates lead to an imaginary-time evolution at the qubit level.

Proposition 7.3
When a gate generator is of the form

$$\hat{H} = \sum_{m,m'=1}^{M} \Delta^{\dagger}_{mm'}\,\hat{a}_m \hat{a}_{m'} + \sum_{m,m'=1}^{M} \Delta_{mm'}\,\hat{a}^{\dagger}_m \hat{a}^{\dagger}_{m'}\,, \tag{7.47}$$

where Δ is a symmetric matrix, then $\{\Omega, K\} = 0$, and the propagator $e^{t\Omega K}$ is an imaginary time evolution under the effective Hamiltonian $-\Omega K$.

Proof. In terms of positions and momenta, we find

$$\begin{aligned}
\hat{H} &= \frac{1}{2} \sum_{m,m'=1}^{2^n} \Delta^{\dagger}_{mm'} \, (q_m + ip_m)(q_{m'} + ip_{m'}) \\
&+ \frac{1}{2} \sum_{m,m'=1}^{2^n} \Delta_{mm'} \, (q_m - ip_m)(q_{m'} - ip_{m'}) \\
&= \frac{1}{2} \sum_{m,m'=1}^{2^n} \Delta^{\dagger}_{mm'} \, (q_m q_{m'} - p_m p_{m'} + i q_m p_{m'} + i p_m q_{m'}) \\
&+ \frac{1}{2} \sum_{m,m'=1}^{2^n} \Delta_{mm'} \, (q_m q_{m'} - p_m p_{m'} - i q_m p_{m'} - i p_m q_{m'}) \\
&= \sum_{m,m'=1}^{2^n} \mathrm{Re}\left[\Delta_{mm'}\right] \, (q_m q_{m'} - p_m p_{m'}) \\
&+ \sum_{m,m'=1}^{2^n} \mathrm{Im}\left[\Delta_{mm'}\right] \, (q_m p_{m'} + p_m q_{m'}) \, .
\end{aligned} \tag{7.48}$$

In this case, the K matrix is a real linear combination of Paulis of the form $Z \otimes P_s$ (corresponding to the first sum in Equation (7.48)) and/or $X \otimes P_s$ (corresponding to the second sum in Equation (7.48)). This implies that $\{\Omega, K\} = 0$. Then, either ΩK is a real combination of Paulis of the form $(iY \otimes I_M)(Z \otimes P_s) = X \otimes P_s$ or $(iY \otimes I_M)(X \otimes P_s) = Z \otimes P_s$ (both of which are in the symplectic algebra according to Proposition 7.1). That is, ΩK is Hermitian and the symplectic propagator $e^{t\Omega K}$ is given by the imaginary-time evolution of the effective gate generator $-\Omega K$. ☐

7.6.2. From bosonic gates to qubit gates

Local gates

- **Phase gate:** This gate is described by the generator $\hat{H} = \hat{q}_m^2 + \hat{p}_m^2$ in terms of bosonic operators. Therefore the real symmetric K matrix can be expressed as

$$\begin{aligned} K &= 2\left(|m\rangle\langle m| + |m+M\rangle\langle m+M|\right) \\ &= 2\left(|0\rangle\langle 0| \otimes |m\rangle\langle m| + |1\rangle\langle 1| \otimes |m\rangle\langle m|\right) \\ &= 2I \otimes |m\rangle\langle m| \; . \end{aligned} \tag{7.49}$$

The associated generator acting on the qubit picture is $\Omega K = 2iY \otimes |m\rangle\langle m|$, and thus

$$\begin{aligned} e^{t\Omega K} &= \sum_{s=0}^{\infty} \frac{(t\Omega K)^s}{s!} = \sum_{s=0}^{\infty} \frac{(2itY \otimes |m\rangle\langle m|)^s}{s!} \\ &= I_{2M} + (\cos(2t) - 1) I \otimes |m\rangle\langle m| + i\sin(2t) Y \otimes |m\rangle\langle m| \\ &= I \otimes \overline{|m\rangle\langle m|} + I \otimes |m\rangle\langle m| \\ &+ (\cos(2t) - 1) I \otimes |m\rangle\langle m| + i\sin(2t) Y \otimes |m\rangle\langle m| \\ &= I \otimes \overline{|m\rangle\langle m|} + (\cos(2t)) I \otimes |m\rangle\langle m| + i\sin(2t) Y \otimes |m\rangle\langle m| \\ &= I \otimes \overline{|m\rangle\langle m|} + e^{2itY} \otimes |m\rangle\langle m| \; . \end{aligned} \tag{7.50}$$

Above we have used the fact that $I_{2M} = I \otimes |m\rangle\langle m| + I \otimes \overline{|m\rangle\langle m|}$ where $\overline{|m\rangle\langle m|} := I - |m\rangle\langle m|$ is the projector onto the orthogonal complement of $|m\rangle$. Hence this gate acts trivially when the state in the register qubits is $|m'\rangle$ such that $m' \neq m$, while it applies an $R_y(4t)$ rotation on the symplectic qubit otherwise (here we assume the standard definition $R_y(\theta) = e^{-i\theta Y/2}$). Hence, the associated qubit gate is a SELECT-$R_y(4t)$.

- **Beamsplitter:** This gate is described by $\hat{H} = \hat{q}_m \hat{p}_{m'} - \hat{q}_{m'} \hat{p}_m$, where $m \neq m'$, in bosonic operators. Therefore the real symmetric K matrix can be expressed as

$$\begin{aligned} K &= 2\left(|m'+M\rangle\langle m| + |m\rangle\langle m'+M| \right. \\ &\left. - |m+M\rangle\langle m'| - |m'\rangle\langle m+M|\right) \\ &= 2iY \otimes \left(|m\rangle\langle m'| - |m'\rangle\langle m|\right) . \end{aligned} \tag{7.51}$$

Therefore $\Omega K = 2iI \otimes (i\,|m\rangle\langle m'| - i\,|m'\rangle\langle m|)$. Here, instead of directly exponentiating this operator and finding a closed formula (as we did with the phase gate), we will derive a sequence of gates whose combined actions lead to $e^{\Omega K}$.

We begin by noting that ΩK acts trivially on the symplectic qubit.

Therefore we focus on the action on the register qubits, where it corresponds to a y rotation in the subspace spanned by $|m\rangle$ and $|m'\rangle$. We write the associated classical bitstrings as $m = m_n \cdots m_1$, and analogously for m'. We denote the bitstring operation $\overline{x}$ as taking the bit-wise negation of each individual bit. We separate the bitstring indices between those where the bits of m and m' match, and those where they differ. We call the first set $e = \{e_j\}_{1\leq j\leq E}$, such that $m_{e_j} = m'_{e_j}\ \forall j$, and the second set $d = \{d_j\}_{1\leq j\leq D}$, such that $m_{d_j} = \overline{m'_{d_j}}\ \forall j$, where $E + D = n$. We can then factorize Equation (7.51) to obtain

$$\Omega K = 2iI \otimes \left(\prod_{j=1}^{E} \left|m_{e_j}\middle\rangle\!\middle\langle m_{e_j}\right|_{e_j} \cdot \left(i\, |m_d\rangle\!\langle \overline{m_d}| - i\, |\overline{m_d}\rangle\!\langle m_d|\right)_d \right) , \tag{7.52}$$

where the notation $\left|m_{e_j}\middle\rangle\!\middle\langle m_{e_j}\right|_{e_j}$ indicates a projector on $|e_j\rangle$ on qubit e_j and identity on the rest, and $(i\, |m_d\rangle\!\langle \overline{m_d}| - i\, |\overline{m_d}\rangle\!\langle m_d|)_d$ acts non-trivially on the qubits whose indexes belong in d. In fact, it is a y-rotation in the subspace spanned by $(|\overline{m_d}\rangle, |m_d\rangle)$. We (arbitrarily) choose to map this rotation to the least-significant qubit where the m and m' bitstrings differ, that is, on qubit d_1. To do so we are going to implement a change of basis using controlled multi-NOTs, so that $|m_d\rangle \to |0\rangle \otimes |0\rangle^{\otimes D-1}$ and $|\overline{m_d}\rangle \to |1\rangle \otimes |0\rangle^{\otimes D-1}$. First, we apply $B := X_{d_1}^{m_{d_1}}$ so that the least-significant qubit matches the previous expression. Then we apply the two following controlled multi-NOT:

$$C_0 := \mathrm{CTRL}(|0\rangle\!\langle 0|_{d_1}) \prod_{j=2}^{D} X_{d_j}^{m_{d_j}} , C_1 := \mathrm{CTRL}(|1\rangle\!\langle 1|_{d_1}) \prod_{j=2}^{D} X_{d_j}^{\overline{m_{d_j}}} , \tag{7.53}$$

where $\mathrm{CTRL}(\Pi)U_k$ is the gate U applied to qubits from set k controlled on the single-qubit local projector Π, for example, $\mathrm{CTRL}(|1\rangle\!\langle 1|_5)X_1$ is an X-gate on qubit 1 controlled by the qubit 5. Then we can apply the SELECT-R_y gate with a target on the qubit whose index is d_1, whose gate generator is as follows.

$$SY := \prod_{j=1}^{E} \left|m_{e_j}\middle\rangle\!\middle\langle m_{e_j}\right|_{e_j} \prod_{j=2}^{D} |0\rangle\!\langle 0|_{d_j} Y_{d_1} . \tag{7.54}$$

Finally, we apply the Hermitian conjugate of the change of basis $B^\dagger C_0^\dagger C_1^\dagger = BC_0C_1$ to return to our original basis. Overall the transformation goes as follows:

$$
\begin{aligned}
e^{2itI\otimes i(|m\rangle\langle m'|-|m'\rangle\langle m|)} &= e^{2itI\otimes C_1C_0BSYBC_0C_1} \\
&= I\otimes C_1C_0Be^{2itSY}BC_0C_1\,, \\
\text{with}\quad e^{2itSY} &= \text{SELECT}\left(\prod_{j=1}^{E}\left|m_{e_j}\right\rangle\!\left\langle m_{e_j}\right|_{e_j}\prod_{j=2}^{D}|0\rangle\!\langle 0|_{d_j}\right)R_y(4t)_{d_1}
\end{aligned}
\tag{7.55}
$$

where $\text{SELECT}(\Pi)U_k$ is the gate U applied to qubits from set k controlled on the projector Π, for example, $\text{SELECT}(|0\rangle\!\langle 0|_3\,|1\rangle\!\langle 1|_5)X_1$ is an X-gate on qubit 1 controlled by the qubit 5 and 0-controlled by qubit 3.

- **Squeezing Gate:** This gate is described by the generator $\hat{H} = \pm(\hat{p}_m\hat{q}_m + \hat{q}_m\hat{p}_m)$. Therefore the real symmetric K matrix can be expressed as

$$
\begin{aligned}
K &= 2(|m\rangle\!\langle m+M| + |m+M\rangle\!\langle m|) && (7.56)\\
&= 2(|0\rangle\!\langle 1| + |1\rangle\!\langle 0|)\otimes|m\rangle\!\langle m| && (7.57)\\
&= 2X\otimes|m\rangle\!\langle m|\;. && (7.58)
\end{aligned}
$$

The associated qubit generator is $\Omega K = \pm 2Z\otimes|m\rangle\!\langle m|$. It is an imaginary time evolution $e^{\pm 2tZ\otimes|m\rangle\langle m|} = e^{\mp it'2Z\otimes|m\rangle\langle m|}$, where $t' = it$, under the effective Hamiltonian ΩK. For small t we have $e^{\pm 2tZ\otimes|m\rangle\langle m|} = I \pm 2tZ\otimes|m\rangle\!\langle m| + O(t^2)$. And therefore we want the state $\left|\hat{\vec{z}}\right\rangle$ to be transformed (up to normalization) as

$$
\begin{aligned}
\left|\hat{\vec{z}}\right\rangle \rightarrow{}& (1\pm 2t)\langle\hat{q}_m\rangle\,|0\rangle\otimes|m\rangle + (1\mp 2t)\langle\hat{p}_m\rangle\,|1\rangle\otimes|m\rangle \\
&+ \sum_{m'\neq m}\langle\hat{q}_{m'}\rangle\,|0\rangle\otimes|m'\rangle + \langle\hat{p}_{m'}\rangle\,|1\rangle\otimes|m'\rangle\;.
\end{aligned}
\tag{7.59}
$$

This can be implemented by a heralded protocol as a Linear Combination of Unitaries (LCU) of the form

$$
\begin{aligned}
&aI + b(I \otimes |m\rangle\langle m| - I \otimes \overline{|m\rangle\langle m|}) \\
&+ c(Z \otimes |m\rangle\langle m| + I \otimes \overline{|m\rangle\langle m|}) + d(-Z \otimes |m\rangle\langle m| + I \otimes \overline{|m\rangle\langle m|}) \,,
\end{aligned}
\tag{7.60}
$$

with $a + b + c + d = 1$ and $a, b, c, d \geq 0$. Applied to the state $\left|\hat{\vec{z}}\right\rangle$ the above LCU yields the following state

$$
\begin{aligned}
&(a + b + c - d)\langle \hat{q}_m \rangle \, |0\rangle \, |m\rangle + (a + b - c + d)\langle \hat{p}_m \rangle \, |1\rangle \, |m\rangle \\
&\qquad + (a - b + c + d) \sum_{m' \neq m} \langle \hat{q}_{m'} \rangle \, |0\rangle \left|m'\right\rangle + \langle \hat{p}_{m'} \rangle \, |1\rangle \left|m'\right\rangle \,.
\end{aligned}
\tag{7.61}
$$

We want this state to be proportional to the one in Equation (7.59) by a factor $\gamma < 1$. We thus need to solve the linear system of equations

$$
\begin{cases}
a + b + c + d & = 1 \\
a + b + c - d & = \gamma(1 \pm 2t) \\
a + b - c + d & = \gamma(1 \mp 2t) \\
a - b + c + d & = \gamma
\end{cases} \,,
\tag{7.62}
$$

whose solution is

$$
a = \frac{3\gamma - 1}{2} \,, \quad b = \frac{1 - \gamma}{2} \,, \quad c = \frac{1 - \gamma \pm 2\gamma t}{2} \,, \quad d = \frac{1 - \gamma \mp 2\gamma t}{2} \,.
\tag{7.63}
$$

Since c and d need to be larger than 0,

$$
\gamma \leq \frac{1}{1 \mp 2t} \,, \qquad \gamma \leq \frac{1}{1 \pm 2t} \,.
\tag{7.64}
$$

In order to maximize the probability of success we should choose the maximum γ that satisfies these constraints. Depending on the sign of $\hat{H}$ one or the other inequalities above is saturated. Therefore we choose $\gamma = 1/(1 + 2t)$. This yields

$$
a = \frac{1 - t}{1 + 2t} \,, \qquad b = \frac{t}{1 + 2t} \,, \qquad c = \frac{t \pm t}{1 + 2t} \,, \qquad d = \frac{t \mp t}{1 + 2t} \,.
\tag{7.65}
$$

Let us calculate the probability of success as the norm of the state

7

in Equation (7.61) for small t,

$$\frac{1}{(1+2t)^2}\left((1\pm 2t)^2\langle q_m\rangle^2+(1\mp 2t)^2\langle p_m\rangle^2+(1-\langle q_m\rangle^2-\langle p_m\rangle^2)\right) \approx \frac{1\pm 4t\langle q_m\rangle^2\mp 4t\langle p_m\rangle^2}{1+4t}\,. \quad (7.66)$$

When $\hat{H}=\hat{p}_m\hat{q}_m+\hat{q}_m\hat{p}_m$, we can see that the best case is $\langle q_m\rangle=1$ and $\langle p_m\rangle=0$, which yields a probability of failure of 0, and the worst case is $\langle p_m\rangle=1$, which yields a probability of failure of $\sim 8t$.

For a squeezing gate, we are given a bitstring description of the mode to which it applies, and the squeezing parameter t. This allows us to compute (a,b,c,d) as per the above equations. In practice, we add two ancillary qubits initialized to zero. We design a unitary U such that $|00\rangle \xrightarrow{U} \sqrt{a}\,|00\rangle+\sqrt{b}\,|01\rangle+\sqrt{c}\,|10\rangle+\sqrt{d}\,|11\rangle$, and apply it to the ancillary register. Then we apply the SELECT-unitaries as per the LCU, derived above. As one unitary is the identity, we do not need to apply this gate, and as either c or d is zero we do not need to apply the corresponding gate either. Each SELECT gate also has controls on the register to select the mode it is being applied to). If the gate generator $\hat{H}$ has a positive sign, then $d=0$, and we apply successively:

$$B \coloneqq \mathrm{SELECT}(|01\rangle\langle 01|_a)(e^{i\pi})_0\mathrm{SELECT}(|m\rangle\langle m|)_r \quad (7.67)$$

$$C \coloneqq \mathrm{SELECT}(|10\rangle\langle 10|_a)Z_0\mathrm{SELECT}(|m\rangle\langle m|)_r\,. \quad (7.68)$$

We denote $O_{a/0/r}$ operations applied to the ancillary/ symplectic/ register qubit(s). We then apply the hermitian conjugate of the state preparation $U^\dagger$, and post-select those states where the ancillary qubits are measured to be the $|00\rangle$ state. Overall the transformation is $|00\rangle\langle 00|_a\, U_a^\dagger BCU_a\, |00\rangle\langle 00|_a$.

- **Displacement gate:** Displacement gates cannot be implemented as qubit gates on a single copy of the input states. Indeed displacement implies adding a number to an amplitude, whereas unitaries acting on a single copy of a state can only multiply amplitudes. While access to multiple copies could in principle be used to implement non-linear transformations [205], we do not consider this setting here.

Global bit-structured gates

In this section, we present a list of global bosonic gates that can be easily translated to qubit gates. In particular, the local interferometric gates map to global qubit gates composed of multi-qubit controlled operations. To mitigate this issue, we can combine local GB gates into global ones, such that their qubit counterparts require fewer multi-qubit controls. We will henceforth refer to the GB gates which effectively translate into local qubit operations as *global bit-structured Gaussian gates*.

- **Phase gate:** It is defined with a binary condition describing which modes the same local gate is applied to. It is given as pairs of indices and binary values. For example, $((1,1),(3,0))$ translates into the binary condition $m_1\overline{m_3} = 1$, which means that the least significant bit should be 1 and the third bit should be 0. For 2^3 modes this implies that the rotation gates apply to modes $001 = 1$ and $011 = 3$. In the corresponding qubit gate, this bitstring condition directly translates into a SELECT on the register. We denote these gates as $P(((k_j, b_j))_j, t)$. Using the same notations as for the local gates, and using 0 as the index for the symplectic qubit, for the example $m_1\overline{m_3} = 1$ we find

  $$P(((1,1)(3,0)),t) \to \mathrm{SELECT}\left(|1\rangle\langle 1|_1\, |0\rangle\langle 0|_3\right) R_y(4t)_0\,. \tag{7.69}$$

  The shorter the bitstring condition of the phase gate is, the more local the operation is in qubits (fewer controls in the SELECT) and the less local it is in the interferometer (more modes are acted upon non-trivially). In the case no bit condition is given, the same rotation gate is applied to all modes and therefore it is simply an R_y on the symplectic qubit.

- **Global Beamsplitter:** It is also defined with a binary condition describing which modes the same local gate is applied to. But as a beamsplitter is a two-mode gate, an additional index l is given to determine how the modes are paired. The l-th bit cannot be part of the bitstring condition. Each mode whose index satisfies the bitstring condition is paired with the one whose index has all bits in common but the l-th one.

  For example, the global bit-structured beamsplitter on 2^3 modes described by $((3,0))$, $l = 1$ is applied to the second half of the modes ($\overline{m_3} = 1$), pairing even modes $0m_20$ with odd modes $0m_21$. Therefore it pairs modes $(000, 001)$ and $(010, 011)$. We denote these gates as

$BS(((k_j, b_j))_j, l, t)$. The example gate may then be written as

$$BS(((3,0)),1,t) \to \text{CTRL}\,(|0\rangle\langle 0|_3)\, R_y(4t)_1\,. \tag{7.70}$$

Note that this particular example is of interest because it corresponds to the 0-controlled-R_y gate that is used extensively in the BQP-completeness proof.

- **Squeezing gate:** The modes to which the squeezing applies are also described as a bitstring condition. It is the same as for the rotation gates but with the squeezing apparatus on the ancillary register instead of the R_y on the symplectic qubit. We denote them as $S(((k_j, b_j))_j, r)$.

Notice that when the binary condition applies to all bits, then we retrieve one local gate from the previous section.

7.6.3. BQP-completeness

Here we provide proof that Problem 1 is BQP-complete. Let us first recall Definition 7.2 and Problem 7.4.

Definition 7.2 (Bit-structured interferometer)
A bit-structured interferometer acting on 2^n *nodes consists of L global beamsplitters, such that each global beamsplitter acts on* 2^{n-1} *modes. A global beamsplitter is specified by two natural numbers,* $k \neq l$*, between 1 and n. The global beamsplitter then acts on all the modes with indices* $\{m\}$ *such that their k-th bit is equal to 0, by applying local beamsplitters between modes with indices* m, m' *that only differ in their l-th bit.*

Problem 7.4
Consider a bit-structured interferometer (see Definition 7.2) acting on 2^n *modes with* $L \in \mathcal{O}(\text{poly}(n))$*, and an input state such that the first mode is displaced in position by a real constant x while the state of the remaining modes is the vacuum. Then, decide whether the expectation value of the position on the first mode at the output of the interferometer is*

$$1.\,\langle \hat{q}_1\rangle > \frac{2}{3}x\,, \qquad \text{or} \qquad 2.\,\langle \hat{q}_1\rangle < \frac{1}{3}x\,,$$

given the promise that either one or the other is true.

We prove our main result by showing that Problem 7.4 reduces to a BQP-complete problem and vice-versa.

Theorem 7.2
Problem 7.4 is BQP-complete.

Proof. We recall the following problem, known to be BQP-complete.

Problem 7.5
Given a uniform family of quantum circuits on n qubits with $J \in \mathcal{O}(\text{poly}(n))$ local gates $\{U_j\}_{1\leq j\leq J}$ taken from a universal gate set $\mathcal{S}$, which are applied to the state $|0\rangle^{\otimes n}$ to produce $|\psi\rangle = \prod_j U_j |0\rangle^{\otimes n}$, decide whether

1. *$|\psi\rangle$ has an overlap larger than 2/3 with $|0\rangle^{\otimes n}$, or*
2. *$|\psi\rangle$ has an overlap smaller than 1/3 with $|0\rangle^{\otimes n}$,*

given the promise that either one or the other is true.

Inclusion in BQP

First, we prove that Problem 7.4 is in BQP by showing that Problem 7.4 can be efficiently reduced to Problem 7.5. We are given access to an algorithm to solve Problem 7.5 and a bit-structured interferometer composed of polynomially many layers of global beamsplitters. The initial state in Problem 7.4 is a tensor product of coherent states such that $\langle \hat{q}_m \rangle = x\delta_{m=1}$ and $\langle \hat{p}_m \rangle = 0$. We consider a quantum circuit over $n+1$ qubits, composed of the symplectic qubit and the n register qubits, as explained in the main text. The initial coherent state then corresponds to an input state $(1, 0, \dots, 0)^T = |0\rangle^{\otimes n+1}$ in the qubit picture.

We recall that a uniform family of quantum circuits is a set of circuits $\{C_n\}$ such that a classical Turing machine can produce a description of C_n on input n in time polynomial in n. In our case, the classical description of the beamsplitter gates is a pair of natural numbers smaller or equal than n for a problem of size n. As such, this description can be efficiently translated to a circuit description using the dictionary we provided in Section 7.6.2, as we know that a single layer of global beamsplitters can be mapped to a 0-controlled-R_y gate on the register (see Definition 7.2). Therefore, we can construct a uniform family of quantum circuits implementing the action in phase space of bit-structured interferometers over 2^n modes.

We use access to the solver of Problem 7.5 with the polynomial-size sequence of 0-controlled-R_y gates corresponding to the global beamsplitters to determine whether the output state has an overlap with $|0\rangle^{\otimes n+1}$ that is $> 2/3$ or $< 1/3$. This directly answers the question of whether the final coherent state has a position expectation value for the first mode

$> 2x/3$ or $< x/3$. We have therefore proved that Problem 7.4 reduces to Problem 7.5, implying that it is in BQP.

As a side note, in Section 7.6.2 we show that a broader class of particle-preserving gates can be simulated efficiently by a quantum computer. Indeed we have mapped each local and bit-structured global interferometric gate to a constant number of multi-controlled qubit gates. Each of the multi-controlled gates can be decomposed into $\mathcal{O}(n)$ two-qubit gates. Therefore any interferometer made of a polynomial number of local or bit-structured global gates over 2^n modes can be simulated efficiently by a polynomial-depth circuit acting on $(n+1)$ qubits.

BQP hardness

Second, we prove that Problem 7.4 is BQP-hard. To do so, we show that Problem 7.5 efficiently reduces to Problem 7.4. As for the inclusion proof, the reduction is based on the fact that the beamsplitters composing a bit-structured interferometer as in Definition 7.2 translate to 0-controlled-R_y rotations between all pairs of register qubits (as proven in Section 7.6.2). The key point for the hardness is that 0-controlled-R_y gates constitute a universal gate set for quantum computation, as stated in the following Lemma (which is a restatement of a result in [197]).

Lemma 7.1
The set of 0-controlled-$R_y(\theta)$ rotation gates with control qubit k and target qubit l, with $1 \leq k \neq l \leq n+2$ and $\theta \in [0, 2\pi]$, applied to the initial state $|0\rangle^{\otimes n+2}$, is universal for quantum computation on n qubits.

Proof. Let us suppose that we have an n-qubit quantum state

$$|\psi\rangle = \sum_{r=0}^{2^n-1} a_r e^{i\theta_r} |r\rangle \ , \tag{7.71}$$

where the a_r and θ_r are real numbers, together with the following universal gate set,

$$R_z(\tau) = \begin{pmatrix} e^{i\tau} & 0 \\ 0 & 1 \end{pmatrix} , \quad R_y(\tau) = \begin{pmatrix} \cos(\tau/2) & -\sin(\tau/2) \\ \sin(\tau/2) & \cos(\tau/2) \end{pmatrix} , \tag{7.72}$$

$$F\left(\frac{\pi}{2}\right) = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} , \tag{7.73}$$

where our $R_z(\tau)$ is equivalent to the standard one, as they only differ by a global phase and a relabeling $\tau \to -\tau$. Moreover, our $F(\pi/2)$ gate can be readily mapped to that in the universal set from Ref. [197], by using single-qubit X gates. Since $R_z(\tau)$ and $R_y(\tau)$ can generate any single-qubit gate, both gate sets are equivalent and hence universal.

Let us furthermore suppose that we have an $(n+1)$-qubit quantum state with real amplitudes,

$$|\phi\rangle = \sum_{r=0}^{2^n-1} a_r \cos\theta_r \, |r\rangle \, |0\rangle + a_r \sin\theta_r \, |r\rangle \, |1\rangle \ . \tag{7.74}$$

Clearly, the states in Equation (7.71) and Equation (7.74) contain the same information. We refer to the extra qubit in $|\phi\rangle$ as the ancilla. The action of the universal gates as in Equation (7.72) on $|\psi\rangle$, such that $|\psi\rangle \to |\psi'\rangle$, will then induce an action $|\phi\rangle \to |\phi'\rangle$. We need to show that this induced action can be efficiently implemented using controlled-R_y rotations that act non-trivially when the control qubit is in the $|0\rangle$ state.

We begin with the R_z gate, whose action on a single-qubit state is

$$R_z(r_0 e^{i\theta_0} \, |0\rangle + r_1 e^{i\theta_1} \, |1\rangle) = r_0 e^{i(\theta_0+\tau)} \, |0\rangle + r_1 e^{i\theta_1} \, |1\rangle \ . \tag{7.75}$$

The induced evolution is

$$\begin{gathered} r_0 \cos\theta_0 \, |0\rangle \, |0\rangle + r_0 \sin\theta_0 \, |0\rangle \, |1\rangle + r_1 \cos\theta_1 \, |1\rangle \, |0\rangle + r_1 \sin\theta_1 \, |1\rangle \, |1\rangle \\ \downarrow \\ r_0 \cos(\theta_0+\tau) \, |0\rangle \, |0\rangle + r_0 \sin(\theta_0+\tau) \, |0\rangle \, |1\rangle + r_1 \cos\theta_1 \, |1\rangle \, |0\rangle + r_1 \sin\theta_1 \, |1\rangle \, |1\rangle \ . \end{gathered} \tag{7.76}$$

This can be achieved by performing a 0-controlled-$R_y(\tau)$ gate where the control is the first qubit (i.e., the qubit on which $R_z(\tau)$ would act) and the target is the ancilla.

Next, let us look at the action of $R_y(\tau)$. To implement this gate, we simply need an additional auxiliary qubit in the $|0\rangle$ state and to apply a 0-controlled-R_y gate conditioned on this extra qubit.

Finally, we have the $F\left(\frac{\pi}{2}\right)$ gate, whose action on a two-qubit state is given by

$$\begin{gathered} r_0 e^{i\theta_0} \, |00\rangle + r_1 e^{i\theta_1} \, |01\rangle + r_2 e^{i\theta_2} \, |10\rangle + r_3 e^{i\theta_3} \, |11\rangle \\ \downarrow \\ r_1 e^{i\theta_1} \, |00\rangle - r_0 e^{i\theta_0} \, |01\rangle + r_2 e^{i\theta_2} \, |10\rangle + r_3 e^{i\theta_3} \, |11\rangle \ . \end{gathered} \tag{7.77}$$

The corresponding induced action is

$$\begin{aligned}&r_0\cos\theta_0\,|000\rangle + r_0\sin\theta_0\,|001\rangle + r_1\cos\theta_1\,|010\rangle + r_1\sin\theta_1\,|011\rangle +\\ &r_2\cos\theta_2\,|100\rangle + r_2\sin\theta_2\,|101\rangle + r_3\cos\theta_3\,|110\rangle + r_3\sin\theta_3\,|111\rangle\\ &\qquad\qquad\qquad\qquad\qquad\qquad\downarrow\\ &r_1\cos\theta_1\,|000\rangle + r_1\sin\theta_1\,|001\rangle - r_0\cos\theta_0\,|010\rangle - r_0\sin\theta_0\,|011\rangle +\\ &r_2\cos\theta_2\,|100\rangle + r_2\sin\theta_2\,|101\rangle + r_3\cos\theta_3\,|110\rangle + r_3\sin\theta_3\,|111\rangle\ . \end{aligned} \tag{7.78}$$

The previous can be achieved by simply applying a 0-controlled-$R_y(\pi)$ in the first two qubits (i.e. the ancilla is not necessary). Therefore, we conclude that the set of 0 is universal for quantum computation, given the initial state $|0\rangle^{\otimes n+2}$. □

We are given a circuit composed of a polynomial number of 0-controlled-R_y gates. We are also given access to a solver for Problem 7.4. We add a qubit on top of the given circuit which is acted trivially upon, and consider it as the symplectic qubit. We query the given solver with the sequence of global bit-structured beamsplitters corresponding to the sequence of 0-controlled-R_y gates as input. Similarly to the inclusion proof, because the input state is the all-zero state, the reduction directly follows. We conclude that Problem 7.4 is BQP-complete. □

7.6.4. From unitary quantum circuits to interferometers

Separating real and imaginary parts of the amplitudes of a quantum state

Consider a unitary quantum circuit on n qubits. Such a circuit is applied to a complex state on n qubits of the following form

$$|\psi\rangle = \sum_{r=0}^{2^n-1} (a_r + b_r i)\,|r\rangle\ . \tag{7.79}$$

Adding one qubit (as the left-most in the tensor product) which we call the symplectic qubit for reasons that will become clear later, we can define the real-valued state over $n+1$ qubits.

$$|\phi\rangle = \sum_{r=0}^{2^n-1} a_r\,|0\rangle\,|r\rangle + b_r\,|1\rangle\,|r\rangle\ . \tag{7.80}$$

First, we recall that a set of universal one-qubit gates, together with any entangling gate forms a universal set. We can use Rz and Ry gates to generate any Rx gate we wish, and thus Rz and Ry form a universal gate set for unitaries. Therefore together with CRy, they form a universal gate set for unitaries on n qubits. This yields the following lemma.

Lemma 7.2
The set of gates $\{Rz, Ry, CRy\}$ *is universal.*

We are going to prove that this universal set of gates $\{Ry, Rz, CRy\}$ for unitary circuits can be translated to a specific set of orthogonal gates on $n+1$ qubits. It is easy to see that for any real gate, such as Ry, the gate is simply applied to the register.

$$Ry(\tau) = \begin{pmatrix} \cos(\tau/2) & -\sin(\tau/2) \\ \sin(\tau/2) & \cos(\tau/2) \end{pmatrix} \rightarrow$$
$$I \otimes Ry(\tau/2) = \begin{pmatrix} \cos(\tau/2) & -\sin(\tau/2) & 0 & 0 \\ \sin(\tau/2) & \cos(\tau/2) & 0 & 0 \\ 0 & 0 & \cos(\tau/2) & -\sin(\tau/2) \\ 0 & 0 & \sin(\tau/2) & \cos(\tau/2) \end{pmatrix}$$
$$= \exp(-i\tau I \otimes Y/2). \quad (7.81)$$

This is also true for controlled-Ry, which yields the same controlled-Ry on the register

$$C_k Ry_l(\tau/2) \rightarrow I \otimes C_k Ry_l(\tau/2) = \exp(-i\tau I \otimes |1\rangle\langle 1|_k \, Y_l/2). \quad (7.82)$$

For complex gates, the symplectic qubit is involved in the corresponding orthogonal gate. We show a derivation for Rz below,

$$Rz(\tau/2)\,|a+ic, b+id\rangle = \begin{pmatrix} 1 & 0 \\ 0 & \cos(\tau/2) + i\sin(\tau/2) \end{pmatrix} \begin{pmatrix} a+ic \\ b+id \end{pmatrix} =$$
$$\begin{pmatrix} a+ic \\ (\cos(\tau/2)b + \sin(\tau/2)d) + i(-\cos(\tau/2)d + \sin(\tau/2)b) \end{pmatrix}. \quad (7.83)$$

Finally we conclude that $Rz(\tau/2) \rightarrow C_1 RY_0(\tau/2) = \exp(-i\tau Y \otimes |k\rangle\langle k|/2)$

$$C_1 RY_0(\tau/2)\,|a,b,c,d\rangle \rightarrow \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\tau/2) & 0 & -\sin(\tau/2) \\ 0 & 0 & 1 & 0 \\ 0 & \sin(\tau/2) & 0 & \cos(\tau/2) \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}. \quad (7.84)$$

We take note that for all of the gates in a universal set, we have derived an orthogonal gate that belongs to the unitary symplectic algebra as in Section 7.6.1, with the generators of $I \otimes Ry \in \{iI \otimes P_a\}$, those of $I \otimes CRy \in \{iI \otimes P_a\}$ and those of $Rz \to C_k Ry_0 \in \{iY \otimes P_s\}$.

Unitary gates to global bit-structured interferometric gates

We now show that each of the gates derived in the previous section maps in turn to a global bit-structured interferometric gate, as follows.

- An Rz gate on qubit k gate is mapped to a $C_k Ry_0$ gate in the symplectic picture. Its corresponding GB gate is the phase gate on half of the modes, whose k-th bit of the index is 1. Using notation from Section 7.6.2 it is $\mathrm{P}((k,1))$.

- An Ry gate on qubit k gate is mapped to a Ry_k gate in the symplectic picture. Its corresponding GB gate is the beamsplitter where all modes are paired such that their index differs only by their k-th qubit. Using notation from Section 7.6.2 it is $\mathrm{BS}(\varnothing, k)$.

- A CRy gate controlled on qubit l and target on qubit k gate is mapped to a $C_l Ry_k$ gate in the symplectic picture. Its corresponding GB gate is the beamsplitter where all the modes whose l-th bit of the index is 1 are paired such that their indices only differ by their k-th qubit. Using notation from Section 7.6.2 it is $\mathrm{BS}((l,1),k)$.

We summarize the previous equivalences in the below table.

Unitary gate	Symplectic gate	GB gate
Rz_k	$C_k Ry_0$	$\mathrm{P}((k,1))$
Ry_k	$I \otimes Ry_k$	$\mathrm{BS}(\varnothing, k)$
$C_l Ry_k$	$I \otimes C_l Ry_k$	$\mathrm{BS}((l,1),k)$

Table 7.1.: Mapping between unitary, symplectic and bosonic gates.

From unitary circuits to interferometers

In this subsection, we show how a unitary qubit computation can be mapped to the evolution by an interferometer of the first moments of expectation values of quadrature operators of coherent states over exponentially many modes.

Gates. Based on the derivations above, given a unitary on n qubits composed of L gates from the universal set $\{Rz, Ry, CRy\}$, we can map it to an equivalent interferometer on 2^n modes composed of exactly L global bit-structured interferometric gates (global bit-structured beamsplitters and global bit-structured phase gates). In that picture, the k-th mode tracks the amplitude of the qubit state on the computational-basis state $|m\rangle$ with the position as the real part and the momentum as the imaginary part.

State preparation. To prepare a sparse qubit state, we start from the vacuum, and each non-zero entry $a + ib$ is position displaced by a and momentum displaced by b for the corresponding mode. Note we have a degree of freedom to upload a state $|\psi\rangle$ onto our modes up to a multiplicative coefficient. For example the state $((1+i)\,|0\rangle - \sqrt{2}\,|1\rangle)/2$ can be prepared as $q_0 = 1, p_0 = 1, q_1 = \sqrt{2}$ but also as $q_0 = 10, p_0 = 10, q_1 = 10\sqrt{2}$. We use the expectation value of the sum of the number operator for each mode $\langle \hat{n}_m \rangle = \frac{1}{2}\langle \sum_m \hat{q}_m^2 + \hat{p}_m^2 \rangle$ to characterize this degree of freedom when encoding qubit states into bosonic states. This expectation value also corresponds to the number of photons $P = \sum_m \langle \hat{n}_m \rangle$ in the circuit. We recall that coherent states are eigenstates of the annihilation operator $\hat{a}\,|\alpha\rangle = \alpha\,|\alpha\rangle$, therefore $\langle \hat{n} \rangle = \langle \hat{a}^\dagger \hat{a} \rangle = |\alpha|^2 = \frac{1}{2}\left(\langle \hat{q} \rangle^2 + \langle \hat{p} \rangle^2\right)$.

Measurements. We consider the photon counting measurement and show that it corresponds to sampling bitstrings from the qubit circuit. The probability of detecting p photons on the m-th mode is a Poissonian distribution $\Pr(p) = \exp(-e_m) e_m^p / p!$ with an average equal to the energy of the mode $e_m = \langle \hat{n}_m \rangle$. We recall that the Poissonian distribution expresses the probability of a given number of events occurring when these events occur independently at a known constant mean rate, which is in that case e_m. Therefore considering that a number of photons $P = \sum_m \langle \hat{n}_m \rangle$ has been injected at the beginning of the interferometer we should get P photons at the output, distributed according to a compound Poissonian distribution where each mode has a rate of occurrence e_m. Effectively we are getting P bitstring samples according to the distribution $[e_0, e_1, \cdots, e_m]$. Recalling that $e_m = \frac{1}{2}\left(\langle \hat{q} \rangle^2 + \langle \hat{p} \rangle^2\right)$, which is effectively proportional to the probability of sampling the bitstring m at the output of the qubit circuit. The more energy injected at the beginning of the circuit the more samples we get.

Now we consider homodyne detection which measures in the basis of $\hat{p}$, $\hat{q}$ or any combination of the two $\hat{x} = \cos\theta \hat{q} + \sin\theta \hat{p}$. This yields a Gaussian distribution centered around $\langle \hat{x} \rangle$, and for coherent states, with variance $1/2$. Effectively this is equivalent to doing a Hadamard test to access either the real part or the imaginary part of the amplitude of a state on the computational basis, which is affected by shot noise. Increasing the energy at the input of the interferometers increases the precision with which we can measure the real and imaginary parts of the amplitude on $|m\rangle$.

7.6.5. Imaginary time evolution is QMA-hard

Reduction

Finding the ground energy of a Hamiltonian is crucial in areas like quantum chemistry and condensed matter physics. This information is often key for understanding quantum phase transitions, chemical reactions or material properties. It can be formulated as a decision problem which is the essence of the class QMA. We recall the QMA-complete ground energy problem below [206].

Problem 7.6
Given a Hamiltonian $H = \sum_{r=1}^{R} H_r$*, with* $R \in O(\mathrm{poly}(n))$*, and each term* H_r *k-local, whose norm* $\|H\| \in O(\mathrm{poly}(n))$*, and two real numbers* $a < b$ *with* $b - a \in \Omega(\mathrm{poly}(n)^{-1})$*, decide whether the ground state energy* λ_0 *is*

1. $\lambda_0 > b$ *or*

2. $\lambda_0 < a$

given that either one is true.

In this appendix, we prove that the ground energy problem, Problem 7.6, can be efficiently reduced to imaginary time evolution, Problem 7.3.

Given a Hamiltonian H on n qubits, a time β and an input state $|x\rangle$, we derive the expectation value of H of the state prepared by the imaginary time evolution. In particular, we are trying to upper bound such a value based on the value of the ground energy λ_0. Then we provide a lower bound on β such that if λ_0 is smaller than a value a with high probability when $|x\rangle$ is taken randomly from a 2-design for each round, which is equivalent to the maximally mixed state

First, we write the eigendecomposition of the Hamiltonian as $H = \sum_k \lambda_k |k\rangle\langle k|$ with λ_0 the ground energy. We also write our input state as $\rho = I/2^n$. The result of the normalised imaginary time evolution is:

$$\rho \to \frac{\sum_k e^{-2\beta\lambda_k} |k\rangle\langle k|}{\sum_k e^{-2\beta\lambda_k}}\,. \tag{7.85}$$

The expectation value of such a state is as follows.

$$\langle H\rangle = \frac{\sum_k \lambda_k e^{-2\beta\lambda_k}}{\sum_k e^{-2\beta\lambda_k}} = \frac{\lambda_0 + \sum_{k>0} \lambda_k e^{-2\beta(\lambda_k-\lambda_0)}}{1+\sum_{k>0} e^{-2\beta(\lambda_k-\lambda_0)}}\,. \tag{7.86}$$

As β grows bigger the term vanishing the least fast in the sum will be the smallest $\lambda_k - \lambda_0$. In the worst case the first excited state is fully degenerate and we have $\lambda_k = \lambda_1, \forall k \geq 1$. Writing the following quantity $A := 2^n - 1$, this yields :

$$\langle H\rangle \leq \frac{\lambda_0 + A\lambda_1 e^{-2\beta(\lambda_1-\lambda_0)}}{1+Ae^{-2\beta(\lambda_1-\lambda_0)}}\,. \tag{7.87}$$

We want to upper bound that quantity for any λ_1, so we consider the function $\lambda_1 \to \frac{\lambda_0 + A\lambda_1 e^{-2\beta(\lambda_1-\lambda_0)}}{1+Ae^{-2\beta(\lambda_1-\lambda_0)}}$, its derivative with respect to λ_1 is proportional (positively) to

$$\begin{aligned}
&A(-2\beta\lambda_1+1)e^{-2\beta(\lambda_1-\lambda_0)}\left(1+Ae^{-2\beta(\lambda_1-\lambda_0)}\right)\\
&-\left(\lambda_0 + A\lambda_1 e^{-2\beta(\lambda_1-\lambda_0)}\right)\left(-2\beta A e^{-2\beta(\lambda_1-\lambda_0)}\right)\\
&= Ae^{-2\beta(\lambda_1-\lambda_0)}\left((-2\beta\lambda_1+1)\left(1+Ae^{-2\beta(\lambda_1-\lambda_0)}\right)\right.\\
&\left.-(\lambda_0+A\lambda_1)\left(e^{-2\beta(\lambda_1-\lambda_0)}\right)(-2\beta)\right)\\
&= Ae^{-2\beta(\lambda_1-\lambda_0)}\left(Ae^{-2\beta(\lambda_1-\lambda_0)}+1-2\beta(\lambda_1-\lambda_0)\right)\,.
\end{aligned} \tag{7.88}$$

The maximum is reached when $2\beta(\lambda_1-\lambda_0) \geq 0$ is equal to the value that solves $Ae^{-x}+1-x=0$, that is $x = 1+W(A/e)$ where W is the Lambert W function (this follows from the identity $W(x)e^{W(x)} = x$). The maximum is then

$$\frac{\lambda_0 + A\lambda_1 e^{-2\beta(\lambda_1-\lambda_0)}}{1+Ae^{-2\beta(\lambda_1-\lambda_0)}} = \frac{\lambda_0+\lambda_1(2\beta(\lambda_1-\lambda_0)-1)}{1+2\beta(\lambda_1-\lambda_0)-1} = \lambda_1 - \frac{1}{2\beta}\,. \tag{7.89}$$

We know that the solution that the Lambert function is such that $\ln(X) - \ln(\ln(x)) \leq W(x) \leq \ln(x)$ in the limit of large x. Therefore we have

$$2\beta(\lambda_1-\lambda_0) = 1+W(A/e) \leq 1+\ln A/e\,, \tag{7.90}$$

$$\lambda_1 \leq \frac{1+\ln A/e}{2\beta} + \lambda_0\,. \tag{7.91}$$

Finally we get

$$\langle H\rangle \leq \frac{\ln A/e}{2\beta} + \lambda_0\,. \tag{7.92}$$

We suppose we can estimate $\langle H\rangle$ as $\hat{h}$ up to an error ε such that with high probability $\hat{h} \geq \langle H\rangle - \varepsilon$. For reasons that will become clear, we choose $\varepsilon = (1-\alpha)(b-a)$ for any constant $0 < \alpha < 1$. Therefore $\varepsilon \in \mathrm{poly}(n)^{-1}$, and for a polynomial error the estimation of the expectation values of H such that $\|H\| \in \mathrm{poly}(n)$ can be done with quantum phase estimation or direct measurements using a polynomial number of calls to the state preparation algorithm. We are looking to find conditions for $\hat{h}$ to be smaller than b when the input state is the maximally mixed state. In order to have $\langle H\rangle \leq b - \varepsilon$, it is sufficient to have

$$\frac{\ln 2^n/e}{2\beta} + a \leq b - \varepsilon \tag{7.93}$$

$$\beta \geq \frac{n\ln(2) - 1}{b - a - \varepsilon} = \frac{n\ln(2) - 1}{\alpha(b-a)} \tag{7.94}$$

Therefore for any Hamiltonian H choosing β as prescribed above we are guaranteed with high probability that given that the ground state energy of H is smaller than a, the H expectation value of the maximall mixed state that has been imaginary time evolved by H is smaller than b. The end-to-end reduction goes as follows: we prepare a state $|x\rangle$ sampled from a 2-design using a polynomial quantum circuit, and we call the solver for imaginary time evolution with parameters $(|x\rangle\,, H, \beta)$ and repeat that the number of times necessary to estimate the H expectation value (QPE or direct measurements) of such a state up to error ε as prescribed above. This number of iterations is polynomial, as the spectral norm of the Hamiltonian is polynomial and the error is inverse polynomial. If the estimated value is greater than b we decide we are in the no case of Problem 7.6 and otherwise we decide yes. With high probability this is a successful protocol. In addition because the gap is inverse polynomial, we know that $\beta \in \Theta(\mathrm{poly}(n))$. This concludes the reduction from the ground energy problem to imaginary time evolution.

In addition, one can see that this sufficient condition on β for imaginary time evolution to be able to resolve a QMA-complete problem, also becomes necessary for a specific Hamiltonian. Consider a Hamiltonian with the following spectrum: one ground energy λ_0, and a first excited state with energy λ_1 with $2^n - 1$ degeneracy, with λ_1 corresponding to the value that maximizes the right hand side of Equation (7.87) for the optimal β.

Mixed time evolutions

Similarly, given a Hamiltonian H on n qubits, a time β and an input state $|x\rangle$, we derive the expectation value of H of the state prepared by mixed time evolution for a time β. In particular, we are trying to upper bound such a value based on the value of the ground energy λ_0. Then we will derive probabilities of such a value to be smaller than a value a when $|x\rangle$ is taken Haar random and the expectation is estimated up to ε error.

First, we write the eigendecomposition of the Hamiltonian as $H = \sum_k \lambda_k |k\rangle\langle k|$ with λ_0 the ground energy. We consider that we add arbitrary oscillatory terms to such a Hamiltonian,

$$A := \sum_k (\lambda_k + i\omega_k) |k\rangle\langle k| \ . \tag{7.95}$$

The result of the normalised time evolution of the maximally mixed state with A is:

$$\frac{\sum_k e^{-\beta\lambda_k} e^{-i\beta\omega_k} |k\rangle\langle k| e^{-\beta\lambda_k} e^{+i\beta\omega_k}}{\sum_k e^{-2\beta\lambda_k}} \ . \tag{7.96}$$

It is easy to see that the complex parts cancel out and that, the expectation value of such a state is as follows,

$$\langle H\rangle = \frac{\sum_k \lambda_k e^{-2\beta\lambda_k}}{\sum_k e^{-2\beta\lambda_k}} \ . \tag{7.97}$$

Which corresponds exactly to the purely imaginary time evolution case. The whole proof follows unchanged, showing that the addition of oscillatory terms doesn't affect the power of the imaginary time evolution.

7.6.6. Exponentially large interferometers equipped with squeezing gates are PostBQP-hard

In Section 7.4.2 we have seen that the imaginary time evolution of $Z \otimes I$ for times $\beta \in \Theta(\mathrm{poly}(n))$ yields the power to perform post-selection. Suppose we are given an real unitary (i.e. orthogonal) U and $\varepsilon = e^{-q(n)}$ with q a polynomial, such that:

$$U|0\rangle = |\phi\rangle = \sqrt{\varepsilon}\,|0\rangle|\phi_0\rangle + \sqrt{1-\varepsilon}\,|1\rangle|\phi_1\rangle \ . \tag{7.98}$$

The task is to post-select on the first register qubit to be $|0\rangle$, which has an exponentially small probability ε of occurring. We use the mapping in Section 7.6.3 to have an interferometer preparing the GB state such that:

$$\langle \hat{q}_{2k} \rangle = \sqrt{\varepsilon}\langle k \mid \phi_0 \rangle \tag{7.99}$$

$$\langle \hat{q}_{2k+1} \rangle = \sqrt{1-\varepsilon}\langle k \mid \phi_1 \rangle \tag{7.100}$$

$$\langle \hat{p}_k \rangle = 0\,. \tag{7.101}$$

Applying a squeezing to all even modes for squeezing parameter β yields the following state,

$$\langle \hat{q}_{2k} \rangle = e^{\beta}\sqrt{\varepsilon}\langle k \mid \phi_0 \rangle \tag{7.102}$$

$$\langle \hat{q}_{2k+1} \rangle = \sqrt{1-\varepsilon}\langle k \mid \phi_1 \rangle \tag{7.103}$$

$$\langle \hat{p}_k \rangle = 0\,. \tag{7.104}$$

Which effectively yields the power of post-selection following arguments from Section 7.4.2.

Definition 7.3 (bit-structured Gaussian Circuit)
This is the family of circuits composed of any circuit as defined in Definition 7.2, followed by a layer of squeezing gates applied to every other mode with squeezing parameter β.

We define the following problem.

Problem 7.7
Consider a bit-structured Gaussian Circuit (see Definition 7.3) acting on 2^n modes with $L \in \mathcal{O}(\mathrm{poly}(n))$ and $\beta \in poly(n)$, and an input state such that the first mode is displaced in position by a real constant x while the state of the remaining modes is the vacuum. Then, decide whether the expectation value of the position on the first mode at the output of the interferometer is

$$1.\, \langle \hat{q}_1 \rangle > \frac{2}{3}x\,, \qquad \text{or} \qquad 2.\, \langle \hat{q}_1 \rangle < \frac{1}{3}x\,,$$

given the promise that either one or the other is true.

And from the previous, we find that the following theorem holds.

Theorem 7.3
Problem 7.7 is Post-BQP-hard.

CHAPTER 8

Conclusion

We summarize our answer to the first research question:

Research Question 1
What are the minimal resources needed to achieve universality with parameterized quantum circuits?

In Chapter 3, we have refined the understanding of the expressivity of Quantum re-uploading models relative to the number of reuploads L. Precisely, we have shown that quantum reuploading models with repeated layers, follow a Gaussian profile with a width proportional to $\sqrt{L}$. This means that while having in theory a frequency support growing linearly with L, in practice it grows quadratically more slowly. Whether this is a bug or a feature depends on the use case: on the one hand, it enhances generalization performance through Lipschitz regularization, on the other hand, it cannot capture function with fine-details.

In Chapter 4, for continuous distribution modeling, we proved that a type of quantum generative model, that we call *expectation-value samplers* can achieve full universality for continuous multivariate distributions. We gave resource bounds in terms of qubits and number of shots, derived from information-theoretic arguments (e.g., variations of the Holevo bound). We showed that such bounds are tight through the existence of a family of circuits that saturate them. Our results highlight a clear trade-off: more qubits reduce measurement cost and vice versa, but minimal resources must scale polynomially in the problem dimension.

We summarize our answer to the second research question:

Research Question 2
How can we leverage our knowledge of parameterized quantum circuits to construct practical provable quantum learning advantages?

In Chapter 5, we formalized a supervised learning task of learning PQC-based functions when gates with unknown parameters are present in the circuit. We showed that this task can yield provable exponential classical-quantum learning advantages under common complexity assumptions. At the core of our proposed quantum learning algorithm is a new subroutine we call *Fourier Coefficient Extraction*, inspired by the Fourier analysis of PQCs. Using trotterization, we connected this task to the more physically relevant task of learning Hamiltonian dynamics, where the input is a classical data set of input-output pairs from time evolutions with unknown parameters. We note that, using the same process, any BQP-complete problem that can be compiled as a PQC can be used to construct a learning task with a learning separation.

We summarize our answer to the third research question:

Research Question 3
How does the computational complexity of bosonic systems relate to that of classical systems and qubit-based systems?

The algorithm we propose in Chapter 6 to solve polynomial differential equations yields a compilation of primitive bosonic gates to approximate the evolution according to the system dynamics. We note that this compilation is exponential in the degree of the polynomial, indicating that non-polynomial equations (such as gravitational dynamics) yield harder simulations. The level of squeezing of the initial bosonic state controls the spread around the initial condition, therefore the more photons are injected in the system, the more precise the simulation is. This uncertainty is particularly critical when simulating chaotic systems.

In Chapter 7, we focus on Gaussian states on exponentially many modes. We define a decision problem that is shown to be BQP-complete when considering only a small subset of interferometers. We prove that squeezing gates are equivalent to imaginary time evolution, and therefore, their addition boosts the complexity to PostBQP-complete.

All provable advantages demonstrated in this thesis, whether for computational or learning tasks, require deep, reliable circuits with high connectivity that are only feasible on fault-tolerant quantum computers. Current Noisy Intermediate-Scale Quantum (NISQ) devices lack the error correction and coherence necessary to execute such algorithms reliably. While they may support proof-of-concept implementations and limited-scale testing, they are insufficient for realizing scalable, useful, and provable quantum advantage. The results presented in this thesis instead characterize the minimal resources and algorithmic structures necessary for achieving such advantages, delineating a clear target for the capabilities of future quantum hardware.

Bibliography

[1] L. K. Grover & A. M. Sengupta. Classical analog of quantum search. *Physical Review A*, **65**, 032319 (2002). Publisher: American Physical Society.

[2] P. W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, **26**, 1484–1509 (1997). Publisher: Society for Industrial and Applied Mathematics.

[3] V. Dunjko & H. J. Briegel. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Reports on Progress in Physics*, **81**, 074001 (2018). Publisher: IOP Publishing.

[4] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii et al. Variational quantum algorithms. *Nature Reviews Physics*, **3**, 625–644 (2021). Number: 9 Publisher: Nature Publishing Group.

[5] J. Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, **2**, 79 (2018). Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.

[6] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik & J. L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, **5**, 4213 (2014). Publisher: Nature Publishing Group.

[7] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li et al. The Variational Quantum Eigensolver: A review of methods and best practices. *Physics Reports*, **986**, 1–128 (2022).

[8] E. Farhi, J. Goldstone & S. Gutmann. A Quantum Approximate Optimization Algorithm (2014).

[9] M. Benedetti, E. Lloyd, S. Sack & M. Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, **4**, 043001 (2019). ArXiv:1906.07682 [quant-ph].

[10] G. Brassard & P. Hoyer. An Exact Quantum Polynomial-Time Algorithm for Simon's Problem. In *Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems*, pages 12–23 (1997). ArXiv:quant-ph/9704027.

[11] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush & H. Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, **9**, 4812 (2018). Publisher: Nature Publishing Group.

[12] D. Gottesman. The Heisenberg Representation of Quantum Computers (1998). ArXiv:quant-ph/9807006 version: 1.

[13] M. L. Goh, M. Larocca, L. Cincio, M. Cerezo & F. Sauvage. Lie-algebraic classical simulations for variational quantum computing (2023). ArXiv:2308.01432.

[14] E. Fontana, M. S. Rudolph, R. Duncan, I. Rungger & C. Cîrstoiu. Classical simulations of noisy variational quantum circuits. *npj Quantum Information*, **11**, 1–12 (2025). Publisher: Nature Publishing Group.

[15] X. Gao & L. Duan. Efficient classical simulation of noisy quantum computation (2018). ArXiv:1810.03176 [quant-ph].

[16] Y. Shao, F. Wei, S. Cheng & Z. Liu. Simulating Noisy Variational Quantum Algorithms: A Polynomial Approach. *Physical Review Letters*, **133**, 120603 (2024). Publisher: American Physical Society.

[17] I. Kerenidis, J. Landman & N. Mathur. Classical and Quantum Algorithms for Orthogonal Neural Networks (2022). ArXiv:2106.07198 [quant-ph].

[18] M. Cerezo, M. Larocca, D. García-Martín, N. L. Diaz, P. Braccia, E. Fontana et al. Does provable absence of barren plateaus imply classical simulability? Or, why we need to rethink

variational quantum computing (2024). ArXiv:2312.09121 [quant-ph, stat].

[19] R. A. Servedio & S. J. Gortler. Quantum versus Classical Learnability (2000). ArXiv:quant-ph/0007036.

[20] Y. Liu, S. Arunachalam & K. Temme. A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics*, **17**, 1013–1017 (2021). Publisher: Nature Publishing Group.

[21] C. Gyurik & V. Dunjko. Exponential separations between classical and quantum learners (2023). ArXiv:2306.16028 [quant-ph].

[22] R. Molteni, C. Gyurik & V. Dunjko. Exponential quantum advantages in learning quantum observables from classical data (2024). ArXiv:2405.02027 [quant-ph].

[23] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, **4**, 251–257 (1991).

[24] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, **2**, 303–314 (1989). Company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 4 Publisher: Springer-Verlag.

[25] A. Pérez-Salinas, D. López-Núñez, A. García-Sáez, P. Forn-Díaz & J. I. Latorre. One qubit as a Universal Approximant. *Physical Review A*, **104**, 012405 (2021). ArXiv:2102.04032 [quant-ph].

[26] M. Schuld, R. Sweke & J. J. Meyer. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physical Review A*, **103**, 032430 (2021). Publisher: American Physical Society.

[27] L. G. Valiant. A theory of the learnable. *Commun. ACM*, **27**, 1134–1142 (1984).

[28] M. J. Kearns & U. Vazirani. *An Introduction to Computational Learning Theory*. The MIT Press (1994). ISBN 978-0-262-27686-3.

[29] S. Lloyd & S. L. Braunstein. Quantum Computation over Continuous Variables. *Physical Review Letters*, **82**, 1784–1787 (1999). Publisher: American Physical Society.

[30] V. Upreti, D. Rudolph & U. Chabaud. Bounding the computational power of bosonic systems (2025). ArXiv:2503.03600 [quant-ph].

[31] E. Bernstein & U. Vazirani. Quantum Complexity Theory. *SIAM Journal on Computing*, **26**, 1411–1473 (1997). Publisher: Society for Industrial and Applied Mathematics.

[32] J. Watrous. Succinct quantum proofs for properties of finite groups. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 537–546 (2000). ISSN: 0272-5428.

[33] C. Marriott & J. Watrous. Quantum Arthur–Merlin games. *computational complexity*, **14**, 122–152 (2005). Company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 2 Publisher: Birkhäuser-Verlag.

[34] S. Aaronson. Quantum computing, postselection, and probabilistic polynomial-time. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **461**, 3473–3482 (2005). Publisher: Royal Society.

[35] A. Barthe & A. Pérez-Salinas. Gradients and frequency profiles of quantum re-uploading models. *Quantum*, **8**, 1523 (2024). Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.

[36] A. Barthe, M. Grossi, S. Vallecorsa, J. Tura & V. Dunjko. Parameterized quantum circuits as universal generative models for continuous multivariate distributions. *npj Quantum Information*, **11**, 121 (2025). Publisher: Nature Publishing Group.

[37] A. Barthe, M. Y. Rad, M. Grossi & V. Dunjko. Quantum Advantage in Learning Quantum Dynamics via Fourier coefficient extraction (2025). ArXiv:2506.17089 [quant-ph].

[38] A. Barthe, M. Grossi, J. Tura & V. Dunjko. Continuous variables quantum algorithm for solving ordinary differential equations. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 2, pages 48–53. IEEE (2023).

[39] A. Barthe, M. Cerezo, A. T. Sornborger, M. Larocca & D. García-Martín. Gate-Based Quantum Simulation of Gaussian Bosonic Circuits on Exponentially Many Modes. *Physical Review Letters*, **134**, 070604 (2025). Publisher: American Physical Society.

[40] M. Sahebi, A. Barthe, Y. Suzuki, Z. Holmes & M. Grossi. On Dequantization of Supervised Quantum Machine Learning via Random Fourier Features (2025). ArXiv:2505.15902 [quant-ph].

[41] A. Pérez-Salinas, M. Y. Rad, A. Barthe & V. Dunjko. Universal approximation of continuous functions with minimal quantum circuits (2025). ArXiv:2411.19152 [quant-ph].

[42] R. Babbush, D. W. Berry, R. Kothari, R. D. Somma & N. Wiebe. Exponential Quantum Speedup in Simulating Coupled Classical Oscillators. *Physical Review X*, **13**, 041041 (2023). Publisher: American Physical Society.

[43] M. A. Nielsen & I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press (2010). ISBN 978-1-139-49548-6.

[44] B. Coyle, D. Mills, V. Danos & E. Kashefi. The Born supremacy: quantum advantage and training of an Ising Born machine. *npj Quantum Information*, **6**, 1–11 (2020). Number: 1 Publisher: Nature Publishing Group.

[45] J.-G. Liu & L. Wang. Differentiable learning of quantum circuit Born machines. *Physical Review A*, **98**, 062324 (2018). Publisher: American Physical Society.

[46] Y. Zhang, R. Wiersema, J. Carrasquilla, L. Cincio & Y. B. Kim. Scalable quantum dynamics compilation via quantum machine learning (2024). ArXiv:2409.16346 [cond-mat, physics:quant-ph].

[47] C. Cîrstoiu, Z. Holmes, J. Iosue, L. Cincio, P. J. Coles & A. Sornborger. Variational fast forwarding for quantum simulation beyond the coherence time. *npj Quantum Information*, **6**, 1–10 (2020). Publisher: Nature Publishing Group.

[48] S. Sim, P. D. Johnson & A. Aspuru-Guzik. Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms. *Advanced Quantum Technologies*, **2**, 1900070 (2019). _eprint: https://advanced.onlinelibrary.wiley.com/doi/pdf/10.1002/qute.201900070.

[49] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster & J. I. Latorre. Data re-uploading for a universal quantum classifier. *Quantum*, **4**, 226 (2020). Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.

[50] S. Jerbi, L. J. Fiderer, H. Poulsen Nautrup, J. M. Kübler, H. J. Briegel & V. Dunjko. Quantum machine learning beyond kernel methods. *Nature Communications*, **14**, 517 (2023). Publisher: Nature Publishing Group.

[51] E. R. Anschuetz & B. T. Kiani. Quantum variational algorithms are swamped with traps. *Nature Communications*, **13**, 7760 (2022). Publisher: Nature Publishing Group.

[52] Z. Holmes, K. Sharma, M. Cerezo & P. J. Coles. Connecting Ansatz Expressibility to Gradient Magnitudes and Barren Plateaus. *PRX Quantum*, **3**, 010313 (2022). Publisher: American Physical Society.

[53] C. Ortiz Marrero, M. Kieferová & N. Wiebe. Entanglement-Induced Barren Plateaus. *PRX Quantum*, **2**, 040316 (2021). Publisher: American Physical Society.

[54] M. Cerezo, A. Sone, T. Volkoff, L. Cincio & P. J. Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications*, **12**, 1791 (2021). Publisher: Nature Publishing Group.

[55] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio & P. J. Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nature Communications*, **12**, 6961 (2021). Publisher: Nature Publishing Group.

[56] A. Arrasmith, M. Cerezo, P. Czarnik, L. Cincio & P. J. Coles. Effect of barren plateaus on gradient-free optimization. *Quantum*, **5**, 558 (2021). Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.

[57] L. Schatzki, M. Larocca, Q. T. Nguyen, F. Sauvage & M. Cerezo. Theoretical guarantees for permutation-equivariant quantum neural networks. *npj Quantum Information*, **10**, 1–14 (2024). Publisher: Nature Publishing Group.

[58] M. Ragone, B. N. Bakalov, F. Sauvage, A. F. Kemper, C. Ortiz Marrero, M. Larocca & M. Cerezo. A Lie algebraic theory of barren plateaus for deep parameterized quantum circuits. *Nature Communications*, **15**, 7172 (2024). Publisher: Nature Publishing Group.

[59] H. Mhiri, R. Puig, S. Lerch, M. S. Rudolph, T. Chotibut, S. Thanasilp & Z. Holmes. A unifying account of warm start guarantees for patches of quantum landscapes (2025). ArXiv:2502.07889 [quant-ph].

[60] A. Uvarov & J. Biamonte. On barren plateaus and cost function locality in variational quantum algorithms. *Journal of Physics A: Mathematical and Theoretical*, **54**, 245301 (2021). ArXiv:2011.10530 [quant-ph].

[61] Z. Liu, L.-W. Yu, L.-M. Duan & D.-L. Deng. Presence and Absence of Barren Plateaus in Tensor-Network Based Machine Learning. *Physical Review Letters*, **129**, 270501 (2022). Publisher: American Physical Society.

[62] H.-M. Rieser, F. Köster & A. P. Raulf. Tensor networks for quantum machine learning. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **479**, 20230218 (2023). Publisher: Royal Society.

[63] R. Sweke, E. Recio-Armengol, S. Jerbi, E. Gil-Fuster, B. Fuller, J. Eisert & J. J. Meyer. Potential and limitations of random Fourier features for dequantizing quantum machine learning. *Quantum*, **9**, 1640 (2025). Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.

[64] R. A. Servedio & S. J. Gortler. Equivalences and Separations Between Quantum and Classical Learnability. *SIAM Journal on Computing*, **33**, 1067–1092 (2004). Publisher: Society for Industrial and Applied Mathematics.

[65] E. R. Anschuetz & X. Gao. Arbitrary Polynomial Separations in Trainable Quantum Machine Learning (2024). ArXiv:2402.08606 [quant-ph].

[66] R. E. Schapire. The strength of weak learnability. *Machine Learning*, **5**, 197–227 (1990). Company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 2 Publisher: Kluwer Academic Publishers.

[67] S. L. Braunstein & P. van Loock. Quantum information with continuous variables. *Quantum information with continuous variables*, **77** (2005).

[68] S. D. Bartlett, B. C. Sanders, S. L. Braunstein & K. Nemoto. Efficient Classical Simulation of Continuous Variable Quantum Information Processes. *Physical Review Letters*, **88**, 097904 (2002). Publisher: American Physical Society.

[69] D. Poulin, A. Qarry, R. Somma & F. Verstraete. Quantum Simulation of Time-Dependent Hamiltonians and the Convenient Illusion of Hilbert Space. *Physical Review Letters*, **106**, 170501 (2011). Publisher: American Physical Society.

[70] M. Oszmaniec & Z. Zimborás. Universal Extensions of Restricted Classes of Quantum Operations. *Physical Review Letters*, **119**, 220502 (2017). Publisher: American Physical Society.

[71] G. Adesso, S. Ragy & A. R. Lee. Continuous Variable Quantum Information: Gaussian States and Beyond. *Open Systems & Information Dynamics*, **21**, 1440001 (2014). Publisher: World Scientific Publishing Co.

[72] J. Kempe, A. Kitaev & O. Regev. The Complexity of the Local Hamiltonian Problem. *SIAM Journal on Computing*, **35**, 1070–1097 (2006). Publisher: Society for Industrial and Applied Mathematics.

[73] D. Gosset & D. Nagaj. Quantum 3-SAT Is QMA$_1$-Complete. *SIAM Journal on Computing*, **45**, 1080–1128 (2016). Publisher: Society for Industrial and Applied Mathematics.

[74] M. Crichigno & T. Kohler. Clique Homology is QMA1-hard (2022). ArXiv:2209.11793 [math-ph, physics:quant-ph].

[75] E. Knill & R. Laflamme. On the Power of One Bit of Quantum Information. *Physical Review Letters*, **81**, 5672–5675 (1998). ArXiv:quant-ph/9802037.

[76] T. Morimae, K. Fujii & J. F. Fitzsimons. On the hardness of classically simulating the one clean qubit model. *Physical Review Letters*, **112** (2014). ArXiv:1312.2496 [quant-ph].

[77] P. W. Shor & S. P. Jordan. Estimating Jones polynomials is a complete problem for one clean qubit (2008). ArXiv:0707.2831 [quant-ph].

[78] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand et al. Noisy intermediate-scale

quantum algorithms. *Reviews of Modern Physics*, **94**, 015004 (2022). Publisher: American Physical Society.

[79] J. R. McClean, M. P. Harrigan, M. Mohseni, N. C. Rubin, Z. Jiang, S. Boixo, V. N. Smelyanskiy, R. Babbush & H. Neven. Low-Depth Mechanisms for Quantum Optimization. *PRX Quantum*, **2**, 030312 (2021). Publisher: American Physical Society.

[80] L. Bittel & M. Kliesch. Training Variational Quantum Algorithms Is NP-Hard. *Physical Review Letters*, **127**, 120502 (2021). Publisher: American Physical Society.

[81] J. R. McClean, J. Romero, R. Babbush & A. Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, **18**, 023023 (2016). Publisher: IOP Publishing.

[82] I. G. Ryabinkin, S. N. Genin & A. F. Izmaylov. Constrained Variational Quantum Eigensolver: Quantum Computer Search Engine in the Fock Space. *Journal of Chemical Theory and Computation*, **15**, 249–255 (2019). Publisher: American Chemical Society.

[83] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová et al. Quantum Chemistry in the Age of Quantum Computing. *Chemical Reviews*, **119**, 10856–10915 (2019). Publisher: American Chemical Society.

[84] Y. Li & S. C. Benjamin. Efficient Variational Quantum Simulator Incorporating Active Error Minimization. *Physical Review X*, **7**, 021050 (2017). Publisher: American Physical Society.

[85] K. Bharti & T. Haug. Quantum-assisted simulator. *Physical Review A*, **104**, 042418 (2021). Publisher: American Physical Society.

[86] S. McArdle, T. Jones, S. Endo, Y. Li, S. C. Benjamin & X. Yuan. Variational ansatz-based quantum simulation of imaginary time evolution. *npj Quantum Information*, **5**, 1–6 (2019). Publisher: Nature Publishing Group.

[87] X. Yuan, S. Endo, Q. Zhao, Y. Li & S. C. Benjamin. Theory of variational quantum simulation. *Quantum*, **3**, 191 (2019). Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.

[88] K. Mitarai, M. Negoro, M. Kitagawa & K. Fujii. Quantum circuit learning. *Physical Review A*, **98**, 032309 (2018). Publisher: American Physical Society.

[89] M. Schuld, A. Bocharov, K. M. Svore & N. Wiebe. Circuit-centric quantum classifiers. *Physical Review A*, **101**, 032308 (2020). Publisher: American Physical Society.

[90] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow & J. M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, **567**, 209–212 (2019). Publisher: Nature Publishing Group.

[91] M. Schuld. Supervised quantum machine learning models are kernel methods (2021). ArXiv:2101.11020 [quant-ph].

[92] J. S. Otterbach, R. Manenti, N. Alidoust, A. Bestwick, M. Block, B. Bloom et al. Unsupervised Machine Learning on a Hybrid Quantum Computer (2017). ArXiv:1712.05771 [quant-ph].

[93] C. Zoufal, A. Lucchi & S. Woerner. Quantum Generative Adversarial Networks for learning and loading random distributions. *npj Quantum Information*, **5**, 1–9 (2019). Number: 1 Publisher: Nature Publishing Group.

[94] C. Zoufal, A. Lucchi & S. Woerner. Variational quantum Boltzmann machines. *Quantum Machine Intelligence*, **3**, 1–15 (2021). Company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 1 Publisher: Springer International Publishing.

[95] P.-L. Dallaire-Demers & N. Killoran. Quantum generative adversarial networks. *Physical Review A*, **98**, 012324 (2018). Publisher: American Physical Society.

[96] M. Schuld & N. Killoran. Quantum Machine Learning in Feature Hilbert Spaces. *Physical Review Letters*, **122**, 040504 (2019). Publisher: American Physical Society.

[97] J. G. Vidal & D. O. Theis. Input Redundancy for Parameterized Quantum Circuits (2020). ArXiv:1901.11434 [quant-ph].

[98] A. Pérez-Salinas, J. Cruz-Martinez, A. A. Alhajri & S. Carrazza. Determining the proton content with a quantum computer.

Physical Review D, **103**, 034027 (2021). Publisher: American Physical Society.

[99] S. Thanasilp, S. Wang, N. A. Nghiem, P. Coles & M. Cerezo. Subtleties in the trainability of quantum machine learning models. *Quantum Machine Intelligence*, **5**, 1–22 (2023). Company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 1 Publisher: Springer International Publishing.

[100] T. Hubregtsen, J. Pichlmeier, P. Stecher & K. Bertels. Evaluation of parameterized quantum circuits: on the relation between classification accuracy, expressibility, and entangling capability. *Quantum Machine Intelligence*, **3**, 1–19 (2021). Company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 1 Publisher: Springer International Publishing.

[101] M. Larocca, P. Czarnik, K. Sharma, G. Muraleedharan, P. J. Coles & M. Cerezo. Diagnosing Barren Plateaus with Tools from Quantum Optimal Control. *Quantum*, **6**, 824 (2022). Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.

[102] M. Larocca, N. Ju, D. García-Martín, P. J. Coles & M. Cerezo. Theory of overparametrization in quantum neural networks. *Nature Computational Science*, **3**, 542–551 (2023). Publisher: Nature Publishing Group.

[103] M. C. Caro, E. Gil-Fuster, J. J. Meyer, J. Eisert & R. Sweke. Encoding-dependent generalization bounds for parametrized quantum circuits. *Quantum*, **5**, 582 (2021). ArXiv:2106.03880 [quant-ph].

[104] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin & H. Weinfurter. Elementary gates for quantum computation. *Physical Review A*, **52**, 3457–3467 (1995). Publisher: American Physical Society.

[105] A. Pérez-Salinas & A. Barthe. github repository: QRU_average (2025). Original-date: 2023-08-01T12:03:38Z.

[106] A. Pérez-Salinas, H. Wang & X. Bonet-Monroig. Analyzing variational quantum landscapes with information content. *npj Quantum Information*, **10**, 1–10 (2024). Publisher: Nature Publishing Group.

[107] I. Olkin & H. Rubin. Multivariate Beta Distributions and Independence Properties of the Wishart Distribution. *The Annals of Mathematical Statistics*, **35**, 261–269 (1964). Publisher: Institute of Mathematical Statistics.

[108] P. Billingsley. *Probability and measure.* John Wiley & Sons (2017).

[109] F. J. Schreiber, J. Eisert & J. J. Meyer. Classical Surrogates for Quantum Learning Models. *Physical Review Letters*, **131**, 100803 (2023). Publisher: American Physical Society.

[110] S. Shin, Y. S. Teo & H. Jeong. Exponential data encoding for quantum supervised learning. *Physical Review A*, **107**, 012422 (2023). Publisher: American Physical Society.

[111] H. Gouk, E. Frank, B. Pfahringer & M. J. Cree. Regularisation of neural networks by enforcing Lipschitz continuity. *Machine Learning*, **110**, 393–416 (2021). Company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 2 Publisher: Springer US.

[112] P. L. Bartlett, D. J. Foster & M. J. Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc. (2017).

[113] E. Peters & M. Schuld. Generalization despite overfitting in quantum machine learning models. *Quantum*, **7**, 1210 (2023). Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.

[114] M. S. Rudolph, E. Fontana, Z. Holmes & L. Cincio. Classical surrogate simulation of quantum systems with LOWESA (2023). ArXiv:2308.09109 [quant-ph].

[115] S. Bravyi, D. Gosset & Y. Liu. Classical simulation of peaked shallow quantum circuits (2023). ArXiv:2309.08405 [quant-ph].

[116] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis & H. Neven. Characterizing quantum supremacy in near-term devices. *Nature Physics*, **14**, 595–600 (2018). Publisher: Nature Publishing Group.

[117] R. W. Bailey. Distributional Identities of Beta and Chi-Squared Variates: A Geometrical Interpretation. *The American Statistician*, **46**, 117–120 (1992). Publisher: ASA Website _eprint: https://www.tandfonline.com/doi/pdf/10.1080/00031305.1992.10475864.

[118] W. Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, **58**, 13–30 (1963). Publisher: ASA Website _eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1963.10500830.

[119] A. Zee. *Quantum field theory in a nutshell*, volume 7. Princeton university press (2010).

[120] user26872. reference for multidimensional gaussian integral.

[121] A. Macaluso, L. Clissa, S. Lodi & C. Sartori. A Variational Algorithm for Quantum Neural Networks. In V. V. Krzhizhanovskaya, G. Závodszky, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos & J. Teixeira, editors, *Computational Science – ICCS 2020*, Lecture Notes in Computer Science, pages 591–604. Springer International Publishing, Cham (2020). ISBN 978-3-030-50433-5.

[122] J. Tian, X. Sun, Y. Du, S. Zhao, Q. Liu, K. Zhang et al. Recent Advances for Quantum Neural Networks in Generative Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **45**, 12321–12340 (2023). Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[123] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy & R. Melko. Quantum Boltzmann Machine. *Physical Review X*, **8**, 021050 (2018). Publisher: American Physical Society.

[124] M. Kieferová & N. Wiebe. Tomography and generative training with quantum Boltzmann machines. *Physical Review A*, **96**, 062327 (2017). Publisher: American Physical Society.

[125] S. Cheng, J. Chen & L. Wang. Information Perspective to Probabilistic Modeling: Boltzmann Machines versus Born Machines. *Entropy*, **20**, 583 (2018). Number: 8 Publisher: Multidisciplinary Digital Publishing Institute.

[126] S. Lloyd & C. Weedbrook. Quantum Generative Adversarial Learning. *Physical Review Letters*, **121**, 040502 (2018). Publisher: American Physical Society.

[127] J. Zeng, Y. Wu, J.-G. Liu, L. Wang & J. Hu. Learning and inference on generative adversarial quantum circuits. *Physical Review A*, **99**, 052306 (2019). Publisher: American Physical Society.

[128] J. Romero & A. Aspuru-Guzik. Variational Quantum Generators: Generative Adversarial Quantum Machine Learning for Continuous Distributions. *Advanced Quantum Technologies*, **4**, 2000003 (2021).

[129] H.-L. Huang, Y. Du, M. Gong, Y. Zhao, Y. Wu, C. Wang et al. Experimental Quantum Generative Adversarial Networks for Image Generation. *Physical Review Applied*, **16**, 024051 (2021). Publisher: American Physical Society.

[130] S. Y. Chang, S. Thanasilp, B. L. Saux, S. Vallecorsa & M. Grossi. Latent Style-based Quantum GAN for high-quality Image Generation (2024). ArXiv:2406.02668 [quant-ph].

[131] C. Bravo-Prieto, J. Baglio, M. Cè, A. Francis, D. M. Grabowska & S. Carrazza. Style-based quantum generative adversarial networks for Monte Carlo events. *Quantum*, **6**, 777 (2022). Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.

[132] P.-Y. Kao, Y.-C. Yang, W.-Y. Chiang, J.-Y. Hsiao, Y. Cao, A. Aliper et al. Exploring the Advantages of Quantum Generative Adversarial Networks in Generative Chemistry. *Journal of Chemical Information and Modeling*, **63**, 3307–3318 (2023). Publisher: American Chemical Society.

[133] S. L. Tsang, M. T. West, S. M. Erfani & M. Usman. Hybrid Quantum-Classical Generative Adversarial Network for High Resolution Image Generation. *IEEE Transactions on Quantum Engineering*, **4**, 1–19 (2023). ArXiv:2212.11614 [quant-ph].

[134] S. Chakrabarti, H. Yiming, T. Li, S. Feizi & X. Wu. Quantum Wasserstein Generative Adversarial Networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox & R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc. (2019).

[135] Z. Qu, W. Chen & P. Tiwari. HQ-DCGAN: Hybrid quantum deep convolutional generative adversarial network approach for ECG generation. *Know.-Based Syst.*, **301** (2024).

[136] R. Shu, X. Xu, M.-H. Yung & W. Cui. Variational Quantum Circuits Enhanced Generative Adversarial Network (2024). ArXiv:2402.01791 [quant-ph].

[137] D. Silver, T. Patel, W. Cutler, A. Ranjan, H. Gandhi & D. Tiwari. MosaiQ: Quantum Generative Adversarial Networks for Image Generation on NISQ Computers (2023). ArXiv:2308.11096 [quant-ph].

[138] A. Ambainis, A. Nayak, A. Ta-Shma & U. Vazirani. Dense quantum coding and quantum finite automata. *Journal of the ACM*, **49**, 496–511 (2002).

[139] E. Gil-Fuster, C. Gyurik, A. Pérez-Salinas & V. Dunjko. On the relation between trainability and dequantization of variational quantum learning models (2024). ArXiv:2406.07072 [quant-ph, stat].

[140] V. I. Bogachev, A. V. Kolesnikov & K. V. Medvedev. Triangular transformations of measures. *Sbornik: Mathematics*, **196**, 309 (2005). Publisher: IOP Publishing.

[141] I. Kobyzev, S. D. Prince & M. A. Brubaker. Normalizing Flows: An Introduction and Review of Current Methods. *IEEE Transactions on Pattern Analysis &amp; Machine Intelligence*, **43**, 3964–3979 (2021). Place: Los Alamitos, CA, USA Publisher: IEEE Computer Society.

[142] C.-W. Huang, D. Krueger, A. Lacoste & A. Courville. Neural Autoregressive Flows. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2078–2087. PMLR (2018). ISSN: 2640-3498.

[143] P. Jaini, K. A. Selby & Y. Yu. Sum of Squares Polynomial Flow (2019). ArXiv:1905.02325 [cs, stat].

[144] C. Villani. *Optimal Transport*, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer, Berlin, Heidelberg (2009). ISBN 978-3-540-71049-3 978-3-540-71050-9.

[145] R. Baire. Sur les fonctions de variables réelles. *Annali di Matematica Pura ed Applicata (1898-1922)*, **3**, 1–123 (1899).

[146] M. Laczkovich. Baire 1 functions. *Real Analysis Exchange*, **9**, 15–28 (1984). Publisher: Michigan State University Press.

[147] H. Lebesgue. Une propriété caractéristique des fonctions de classe 1. *Bulletin de la Société Mathématique de France*, **32**, 229–242 (1904). Publisher: Société mathématique de France.

[148] M. Plesch & C. Brukner. Quantum-state preparation with universal gate decompositions. *Physical Review A*, **83**, 032302 (2011). Publisher: American Physical Society.

[149] S. Aaronson. The learnability of quantum states. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **463**, 3089–3114 (2007). Publisher: Royal Society.

[150] S. Jerbi, C. Gyurik, S. C. Marshall, R. Molteni & V. Dunjko. Shadows of quantum machine learning (2023). ArXiv:2306.00061 [quant-ph, stat].

[151] L. Ambrosio, N. Gigli & G. Savaré. *Gradient Flows*. Lectures in Mathematics ETH Zürich. Birkhäuser-Verlag, Basel (2005). ISBN 978-3-7643-2428-5.

[152] D. Xiu. *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton University Press (2010). ISBN 978-0-691-14212-8.

[153] H.-Y. Huang, R. Kueng, G. Torlai, V. V. Albert & J. Preskill. Provably efficient machine learning for quantum many-body problems (2022). ArXiv:2106.12627.

[154] M. C. Caro, H.-Y. Huang, N. Ezzell, J. Gibbs, A. T. Sornborger, L. Cincio, P. J. Coles & Z. Holmes. Out-of-distribution generalization for learning quantum dynamics. *Nature Communications*, **14**, 3751 (2023). ArXiv:2204.10268 [quant-ph].

[155] N. Wiebe, C. Granade, C. Ferrie & D. G. Cory. Hamiltonian Learning and Certification Using Quantum Resources. *Physical Review Letters*, **112**, 190501 (2014). ArXiv:1309.0876 [quant-ph].

[156] M. Schuld, I. Sinayskiy & F. Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, **56**, 172–185 (2015). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00107514.2014.964942.

[157] M. C. Caro, H.-Y. Huang, M. Cerezo, K. Sharma, A. Sornborger, L. Cincio & P. J. Coles. Generalization in quantum

machine learning from few training data. *Nature Communications*, **13**, 4919 (2022). Publisher: Nature Publishing Group.

[158] A. M. Childs. *Quantum information processing in continuous time.* Thesis, Massachusetts Institute of Technology (2004). Accepted: 2005-05-17T14:50:52Z.

[159] S. Arunachalam, A. B. Grilo & A. Sundaram. Quantum hardness of learning shallow classical circuits (2019). ArXiv:1903.02840 [quant-ph].

[160] S. Thanasilp, S. Wang, M. Cerezo & Z. Holmes. Exponential concentration in quantum kernel methods. *Nature Communications*, **15**, 5200 (2024). Publisher: Nature Publishing Group.

[161] G. H. Low & I. L. Chuang. Hamiltonian Simulation by Qubitization. *Quantum*, **3**, 163 (2019). ArXiv:1610.06546 [quant-ph].

[162] N. Wiebe & A. Childs. Hamiltonian Simulation Using Linear Combinations of Unitary Operations. volume 2012, page T30.003 (2012). ADS Bibcode: 2012APS..MART30003W.

[163] M. Christ. A sharpened Hausdorff-Young inequality (2014). ArXiv:1406.1210 [math].

[164] H. Yamasaki, N. Isogai & M. Murao. Advantage of Quantum Machine Learning from General Computational Advantages (2023). ArXiv:2312.03057.

[165] M. Mohri. Foundations of machine learning (2018).

[166] P. K. Faehrmann, J. Eisert & R. Kueng. In the shadow of the Hadamard test: Using the garbage state for good and further modifications (2025). ArXiv:2505.15913 [quant-ph].

[167] C. S. Ong, X. Mary, S. Canu & A. J. Smola. Learning with non-positive kernels. In *Twenty-first international conference on Machine learning - ICML '04*, page 81. ACM Press, Banff, Alberta, Canada (2004).

[168] C. Cortes, M. Mohri & A. Talwalkar. On the impact of kernel approximation on learning accuracy. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 113–120. JMLR Workshop and Conference Proceedings (2010).

[169] O. Bousquet & A. Elisseeff. Stability and generalization. *J. Mach. Learn. Res.*, **2**, 499–526 (2002).

[170] S. Thabet, L. Monbroussou, E. Z. Mamon & J. Landman. When Quantum and Classical Models Disagree: Learning Beyond Minimum Norm Least Square (2024). ArXiv:2411.04940 [quant-ph].

[171] A. W. Harrow, A. Hassidim & S. Lloyd. Quantum Algorithm for Linear Systems of Equations. *Physical Review Letters*, **103**, 150502 (2009). Publisher: American Physical Society.

[172] J.-P. Liu, H. O. Kolden, H. K. Krovi, N. F. Loureiro, K. Trivisa & A. M. Childs. Efficient quantum algorithm for dissipative nonlinear differential equations. *Proceedings of the National Academy of Sciences*, **118**, e2026805118 (2021).

[173] S. Aaronson. Read the fine print. *Nature Physics*, **11**, 291–293 (2015). Number: 4 Publisher: Nature Publishing Group.

[174] D. An, J.-P. Liu, D. Wang & Q. Zhao. A theory of quantum differential equation solvers: limitations and fast-forwarding (2023). ArXiv:2211.05246 [quant-ph].

[175] S. Jin, N. Liu & Y. Yu. Quantum simulation of partial differential equations via Schrodingerisation (2022). ArXiv:2212.13969 [quant-ph].

[176] I. Joseph. Koopman–von Neumann approach to quantum simulation of nonlinear classical dynamics. *Physical Review Research*, **2**, 043102 (2020). Publisher: American Physical Society.

[177] K. Kowalski. Methods of Hilbert Spaces in the Theory of Nonlinear Dynamical Systems. WORLD SCIENTIFIC (1994). ISBN 978-981-02-1753-2 978-981-4354-12-7.

[178] S. Buck, R. Coleman & H. Sargsyan. Continuous Variable Quantum Algorithms: an Introduction (2021). ArXiv:2107.02151 [quant-ph].

[179] A. Engel, G. Smith & S. E. Parker. Linear embedding of nonlinear dynamical systems and prospects for efficient quantum algorithms. *Physics of Plasmas*, **28**, 062305 (2021).

[180] S. Jin, N. Liu & Y. Yu. Time complexity analysis of quantum difference methods for linear high dimensional and multiscale partial differential equations. *Journal of Computational Physics*, **471**, 111641 (2022).

[181] J. Provazník, R. Filip & P. Marek. Taming numerical errors in simulations of continuous variable non-Gaussian state preparation. *Scientific Reports*, **12**, 16574 (2022).

[182] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe & S. Zhu. A Theory of Trotter Error. *Physical Review X*, **11**, 011020 (2021). ArXiv:1912.08854 [cond-mat, physics:physics, physics:quant-ph].

[183] U. Chabaud, D. Markham & F. Grosshans. Stellar Representation of Non-Gaussian Quantum States. *Physical Review Letters*, **124**, 063605 (2020).

[184] A. Winter. Energy-constrained diamond norm with applications to the uniform continuity of continuous variable channel capacities (2017). ArXiv:1712.10267 [math-ph, physics:quant-ph].

[185] S. Aaronson. BQP and the polynomial hierarchy. In *Proceedings of the forty-second ACM symposium on Theory of computing*, STOC '10, pages 141–150. Association for Computing Machinery, New York, NY, USA (2010). ISBN 978-1-4503-0050-6.

[186] S. P. Jordan, H. Krovi, K. S. M. Lee & J. Preskill. BQP-completeness of scattering in scalar quantum field theory. *Quantum*, **2**, 44 (2018). Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.

[187] E. Knill, R. Laflamme & G. J. Milburn. A scheme for efficient quantum computation with linear optics. *Nature*, **409**, 46–52 (2001). Publisher: Nature Publishing Group.

[188] A. Bouland & S. Aaronson. Generation of universal linear optics by any beam splitter. *Physical Review A*, **89**, 062316 (2014). Publisher: American Physical Society.

[189] A. Sawicki. Universality of beamsplitters (2016). ArXiv:1507.08255 [math-ph, physics:quant-ph].

[190] D. García-Martín, P. Braccia & M. Cerezo. Architectures and random properties of symplectic quantum circuits (2024). ArXiv:2405.10264 [quant-ph].

[191] M. Motta, C. Sun, A. T. K. Tan, M. J. O'Rourke, E. Ye, A. J. Minnich, F. G. S. L. Brandão & G. K.-L. Chan. Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution. *Nature Physics*, **16**, 205–210 (2020). Publisher: Nature Publishing Group.

[192] C. Zoufal, D. Sutter & S. Woerner. Error bounds for variational quantum time evolution. *Physical Review Applied*, **20**, 044059 (2023). Publisher: American Physical Society.

[193] T. L. Silva, M. M. Taddei, S. Carrazza & L. Aolita. Fragmented imaginary-time evolution for early-stage quantum signal processors. *Scientific Reports*, **13**, 18258 (2023). Publisher: Nature Publishing Group.

[194] A. M. Childs & N. Wiebe. Hamiltonian Simulation Using Linear Combinations of Unitary Operations. *Quantum Information and Computation*, **12** (2012). ArXiv:1202.5822 [quant-ph].

[195] S. Efthymiou, S. Ramos-Calderer, C. Bravo-Prieto, A. Pérez-Salinas, D. García-Martín, A. Garcia-Saez, J. I. Latorre & S. Carrazza. Qibo: a framework for quantum simulation with hardware acceleration. *Quantum Science and Technology*, **7**, 015018 (2021). Publisher: IOP Publishing.

[196] S. Efthymiou, M. Lazzarin, A. Pasquale & S. Carrazza. Quantum simulation with just-in-time compilation. *Quantum*, **6**, 814 (2022). Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.

[197] T. Rudolph & L. Grover. A 2 rebit gate universal for quantum computing (2002). ArXiv:quant-ph/0210187.

[198] Y. Shi. Both Toffoli and Controlled-NOT need little help to do universal quantum computation (2002). ArXiv:quant-ph/0205115.

[199] D. Aharonov. A Simple Proof that Toffoli and Hadamard are Quantum Universal (2003). ArXiv:quant-ph/0301040.

[200] M. Stroeks, D. Lenterman, B. Terhal & Y. Herasymenko. Solving Free Fermion Problems on a Quantum Computer (2024).

[201] S. K. Leyton & T. J. Osborne. A quantum algorithm to solve nonlinear differential equations (2008). ArXiv:0812.4423 [quant-ph].

[202] D. W. Berry. High-order quantum algorithm for solving linear differential equations. *Journal of Physics A: Mathematical and Theoretical*, **47**, 105301 (2014). ArXiv:1010.2745 [quant-ph].

[203] H. Krovi. Improved quantum algorithms for linear and nonlinear differential equations. *Quantum*, **7**, 913 (2023). ArXiv:2202.01054 [quant-ph].

[204] J. D. Biamonte & P. J. Love. Realizable Hamiltonians for universal adiabatic quantum computers. *Physical Review A*, **78**, 012352 (2008). Publisher: American Physical Society.

[205] Z. Holmes, N. J. Coble, A. T. Sornborger & Y. Subaşı. Nonlinear transformations in quantum computation. *Physical Review Research*, **5**, 013105 (2023). Publisher: American Physical Society.

[206] A. D. Bookatz. QMA-complete problems. *Quantum Information and Computation*, **14**, 361–383 (2014).

Acknowledgments

When I left a steady and comfortable position at the European Space Agency to start a PhD in a field largely unrelated to my previous work, it raised a few eyebrows, but I was fortunate to have the understanding and support of my friends and family. This is a simple thank-you to the people who made the journey both possible and unforgettable.

Vedran and Jordi, thank you for guiding me through this PhD and for teaching me how to think and work as a researcher. This was no small task, given that I arrived thoroughly trained as an engineer. I learned a great deal alongside you and genuinely enjoyed our scientific collaboration. Thank you to Adrian and Patrick for your support, both as postdocs and as friends. And thank you to all the aQa members: even though many of our interactions were two-dimensional, you made me feel part of the group.

On the CERN side, I would like to thank Sofia and Michele for giving me the opportunity to start this PhD. Thank you as well to my office-mates, for sharing the everyday ups and downs of this experience.

I am grateful to the Los Alamos School of Quantum Computing: it gave me not only a paper I am proud of, but also an unforgettable summer, inspiring mentors, and lasting friendships.

Thanks also to the many people I met at conferences, with whom I shared ideas, beers, or both.

Finally, to my friends: each of you, in your own way, makes me feel at home. Listing names would be too long, but I warmly thank my friends from my childhood, and from Lyon, Toulouse, Antibes, Leiden, Rome, Geneva and of course my family.

Samenvatting

Kwantumrekenen introduceert een nieuw rekenmodel dat de principes van de kwantummechanica benut om taken uit te voeren die naar verwachting niet efficiënt oplosbaar zijn voor klassieke computers. Dit proefschrift onderzoekt hoe deze ontwikkeling leeralgoritmen en de simulatie van klassieke fysische systemen kan beïnvloeden. De overkoepelende doelstelling is situaties te identificeren waarin kwantumcomputers een *bewijsbaar* voordeel bieden voor deterministische taken of leertaken.

Het eerste hoofdstuk plaatst het werk in de bredere context van de kwantuminformatica en formuleert drie leidende vragen: (i) wat zijn de minimale hulpbronnen voor universaliteit in *geparameteriseerde kwantummodellen*; (ii) hoe construeer je bewijsbare kwantum-leervoordelen; en (iii) wat is de computationele complexiteit van bosonische systemen en hoe verhouden die zich tot klassieke en qubit-gebaseerde computatie.

Het tweede hoofdstuk biedt achtergrond over kwantumrekenen, statistische leertheorie, bosonische Hamiltonianen, *Parameterized Quantum Circuits* (PQC's) en kwantumcomplexiteitsklassen.

In Hoofdstuk 3 bestuderen we *quantum data re-uploading* (QRU) modellen, d.w.z. *Parameterized Quantum Circuits* die de invoer herhaald herencoderen. We analyseren hun expressiviteit via het frequentiespectrum van de gerealiseerde functies: naarmate het aantal herencoderingen L toeneemt, concentreert het gemiddelde spectrum zich rond een Gaussische verdeling met breedte $\propto \sqrt{L}$, terwijl de support lineair groeit met L. In de praktijk induceert dit een inductieve bias naar gladdere functies—gunstig voor generalisatie, maar beperkend voor fijne details.

In Hoofdstuk 4 verschuiven we van begeleid leren naar generatieve modellering en introduceren *Expectation Value Samplers* (EVS): gegeven een circuit $U(x)$ met willekeurige klassieke invoer x en een verzameling observabelen is de uitvoer de vector van hun verwachtingswaarden. We bewijzen universaliteit op de hyperkubus $[-1,1]^M$ in twee regimes: (i) observabelen met constante spectrale norm op M qubits; (ii) $\Theta(\log M)$ qubits met observabelen van norm $\Theta(M)$. Deze resultaten expliciteren de hulpbronnen-afruil tussen het aantal qubits en de norm van de observabelen, wat direct doorwerkt in de shotcomplexiteit. We laten ook zien hoe EVS natuurlijk aansluit bij veeldeeltjesfysica en in welke situaties dit model passend is.

In Hoofdstuk 5 presenteren we een supervisietaak waarop een kwantumleerder aantoonbaar elke klassieke leerder overtreft onder standaard complexiteitsaannames. De sleutel is een routine voor *Fourier-coëfficiënt-extractie* die spectrale componenten van door PQC's geïnduceerde functies uitleest. Dit levert een feature-map op die klassiek moeilijk te berekenen is en leidt tot een PAC-scheiding. Via Trotterisatie vertalen we hetzelfde idee naar het *leren van Hamiltoniaanse dynamica* uit input-outputparen: efficiënt op een kwantumapparaat, maar klassiek moeilijk. De feature-map induceert tevens een kernel waarvan de Grammatrix efficiënt op een kwantumcomputer kan worden geëvalueerd.

In Hoofdstuk 6 ontwerpen we een kwantumalgoritme met continue variabelen (CV) voor gewone differentiaalvergelijkingen (ODE) binnen het *Koopman–von Neumann*-formalisme (KvN). KvN embedt een klassieke ODE als Hamiltoniaanse evolutie op een oneindig-dimensionale Hilbertruimte. Dit kader is van nature geschikt voor *initiële-distributieproblemen*, d.w.z. de evolutie van een volledige waarschijnlijkheidsdichtheid in plaats van individuele trajecten. We leiden de structuurconstanten van een relevante Lie-algebra af en compileren de Hamiltoniaanse evolutie met hogere-orde Trotterformules tot slechts drie poorttypen. Ten slotte tonen we hoe onzekerheid in de beginvoorwaarden correspondeert met het *squeezing*-niveau van de initiële bosonische toestand.

In Hoofdstuk 7 onderzoeken we de complexiteit van het simuleren van grote *Gaussiaanse bosonische circuits*. Beperking tot een eenvoudige familie interferometers levert een beslissingsprobleem dat BQP-compleet is. Het toevoegen van gestructureerde *squeezing* verhoogt de rekenkracht en maakt het probleem $\mathsf{PostBQP}$-moeilijk. Intuïtief kan *squeezing* samen met eenvoudige interferometrie postselectie-achtige effecten op verwachtingswaarden emuleren, wat de sprong in complexiteit verklaart. Dit hoofdstuk verdiept zo het begrip van de relatie tussen de complexiteit van het simuleren van exponentieel grote klassieke systemen en die van kwantumrekenen.

Summary

Quantum computing offers a new model of computation. The goal is to use the principles of quantum mechanics to perform tasks that are expected to be intractable for classical computers. This thesis investigates how its development could impact learning algorithms and the simulation of classical physical systems. The overarching goal is to identify settings in which quantum computers provide a *provable* computational or learning advantage.

The first chapter introduces this work within the context of quantum computing and introduces three guiding questions. The first concerns the minimal resources required for universality in parameterized quantum models; the second asks how to construct provable quantum learning advantages; and the third examines the computational complexity of bosonic systems and their relation to classical and qubit-based computation.

The second chapter constitutes background on the notions of quantum computation, statistical learning theory, bosonic Hamiltonians, parameterized quantum circuits, and quantum complexity classes.

In Chapter 3, we study *quantum re-uploading models* (QRUs), i.e., parameterized circuits that repeatedly encode the input data. We analyze their expressivity, looking at the frequency spectrum of the functions realized by QRUs: as the number of re-uploads L grows, the average spectrum concentrates into a Gaussian with width scaling like $\sqrt{L}$, while the support grows like L. Practically, QRUs present an inductive bias for smooth functions, which is good for generalization but are limited in capturing fine details.

In Chapter 4, we move from supervised learning to generative modeling and introduce *expectation value samplers* (EVS): given a circuit $U(x)$ with random classical inputs x and a set of observables, the output is the vector of their expectation values. We prove the universality of the EVS model

on the hypercube $[-1,1]^M$ in two regimes: (i) with observables of constant spectral norm on M qubits (ii) with $\Theta(\log M)$ qubits and $\Theta(M)$-norm observables. These theorems make explicit the resource trade-off between qubit count and observable norm, which directly translates into shot complexity. We also show how EVS connects naturally to a many-body physics setting, highlighting when the model is a good fit for physical tasks.

In Chapter 5, we give a supervised learning task where a quantum learner provably outperforms any classical learner under standard complexity assumptions. The key ingredient is a *Fourier coefficient extraction* routine that reads spectral components of PQC-induced functions. This yields a feature map that is hard to compute classically, leading to a PAC learning separation. By Trotterization, we translate the same idea to *learning Hamiltonian dynamics* from input–output pairs: efficient on a quantum device, hard classically. This feature map also yields a kernel whose Gram matrix can be evaluated efficiently on a quantum computer.

In Chapter 6, we design a continuous-variable (CV) quantum algorithm for ordinary differential equations within the Koopman–von Neumann (KvN) framework. KvN embeds a classical ODE as a Hamiltonian evolution on an infinite-dimensional Hilbert space. This framework is naturally suited for *initial distribution problems*, that is, the evolution of a whole probability density rather than single-trajectory initial value problems. The chapter derives the structure constants of a relevant Lie algebra. Using higher-order Trotter formulas, the Hamiltonian evolution is compiled using just three gate types. We show how uncertainty in initial conditions translates into the squeezing level of the initial bosonic state.

In Chapter 7, we study the complexity of simulating large Gaussian bosonic circuits. Restricting to a simple family of interferometers yields a decision problem that we prove is BQP-complete. Adding structured squeezing boosts the computational power to PostBQP-hardness. Intuitively, squeezing plus simple interferometry can emulate postselection-like effects on expectation values, which explains the jump in complexity. This chapter further strengthens our understanding of how the computational complexity of simulating exponentially large classical systems relates to that of quantum computation.

Résumé

L'informatique quantique est un sous-domaine de l'informatique qui traite des ordinateurs quantiques, un nouveau type d'ordinateur. L'enjeu est d'utiliser les principes de la mécanique quantique pour réaliser des calculs infaisables en un temps raisonnable pour des ordinateurs classiques. Ce manuscrit étudie l'impact potentiel de ces technologies sur les algorithmes d'apprentissage et sur la simulation de systèmes physiques classiques. L'objectif est d'identifier des tâches, déterministes ou d'apprentissage, pour lesquelles les ordinateurs quantiques fournissent un avantage prouvable.

Le premier chapitre introduit ce travail dans le contexte plus général de l'informatique quantique et introduit trois questions directrices. La première concerne les ressources minimales requises pour l'universalité dans les *modèles quantiques paramétrés* ; la deuxième demande comment construire des avantages prouvables pour des tâches d'apprentissage ; la troisième examine la complexité computationnelle des systèmes bosoniques et leur relation aux calculs fondés sur des qubits.

Le second chapitre constitue un rappel des notions de calcul quantique, de théorie de l'apprentissage statistique, des Hamiltoniens bosoniques, des *circuits quantiques paramétrés* (PQC) et des classes de complexité quantique.

Le Chapitre 3 traite des *Quantum Re-Uploading models* (QRU), c'est-à-dire des *circuits quantiques paramétrés* qui encodent les données de manière répétée. Nous analysons leur expressivité à travers le spectre fréquentiel des fonctions qu'ils réalisent : lorsque le nombre de ré-encodages L croît, le spectre moyen se concentre en une gaussienne de largeur $\propto \sqrt{L}$, tandis que le support croît comme L. En pratique, les QRU induisent un biais inductif vers des fonctions plus régulières, ce qui permet une meilleure généralisation, mais empêche de capturer des détails fins.

Au Chapitre 4, nous passons de l'apprentissage supervisé à la modéli-

sation générative et introduisons les *Expectation Value Samplers* (EVS) : étant donné un circuit $U(x)$ avec des entrées classiques aléatoires x et un ensemble d'observables, la sortie est le vecteur de leurs valeurs d'espérance. Nous prouvons leur universalité sur l'hypercube $[-1, 1]^M$ dans deux régimes : (i) avec des observables de norme spectrale constante sur M *qubits* (ii) avec $\Theta(\log M)$ qubits et des observables de norme $\Theta(M)$. Ces résultats rendent explicite le compromis de ressources entre nombre de qubits et le nombre d'échantillons.

Au Chapitre 5, nous construisons une tâche d'apprentissage supervisé où un modèle quantique surpasse de manière prouvable tout modèle classique sous des hypothèses de complexité standard. L'ingrédient clé est une routine d'*Fourier coefficient extraction* qui permet d'extraire les composantes spectrales des fonctions induites par des PQC. Elle fournit une *feature map* difficile à calculer classiquement, menant à une séparation d'apprentissage. La même idée s'applique à l'apprentissage de dynamiques hamiltoniennes. Cette *feature map* induit aussi un *kernel* dont la matrice de Gram peut être évaluée efficacement sur un ordinateur quantique.

Au Chapitre 6, nous concevons un algorithme quantique à variables continues pour des équations différentielles ordinaires dans le formalisme de *Koopman–von Neumann* (KvN). KvN transforme une équation différentielle ordinaire non linéaire en une évolution hamiltonienne dans un espace de Hilbert de dimension infinie. Ce cadre se prête naturellement à l'évolution de distributions initiales, c'est-à-dire l'évolution d'une densité de probabilité complète plutôt que de trajectoires individuelles. Le chapitre dérive les constantes de structure de l'algèbre de Lie associée. À l'aide de formules de Trotter d'ordre supérieur, nous compilons l'évolution hamiltonienne en n'utilisant que trois types de portes quantiques.

Au Chapitre 7, nous étudions la complexité de la simulation de grands *circuits bosoniques gaussiens*. La restriction à une famille simple d'interféromètres mène à un problème de décision que nous montrons BQP-complet. L'ajout d'une compression de quadrature structurée augmente la puissance de calcul jusqu'à la $\mathsf{PostBQP}$-difficulté. Intuitivement, la combinaison de compression de quadrature et d'une interférométrie simple peut émuler des effets de type post-sélection sur des valeurs d'espérance, ce qui explique le saut de complexité. Ce chapitre renforce notre compréhension de la manière dont la complexité de calcul de la simulation de systèmes classiques exponentiellement grands se rapporte à celle du calcul quantique.

Curriculum Vitae

I was born on the 28th of September 1993 in Suresnes, France. From a young age, and encouraged by my parents, I cultivated a deep curiosity for science, with a particular fascination for space. I followed a straightforward academic path through a scientific high school, preparatory classes, and finally an Aerospace Engineering degree at ISAE-Supaero, from which I graduated in 2016 after a gap year that brought me to Japan and Australia.

I began my career as an engineer in the space and defense industry. My first professional experience was an internship at the European Space Agency (ESA) in the Netherlands, which also marked my first encounter with Leiden. I then worked at Airbus Defence and Space in Stevenage, UK, followed by Thales in Sophia Antipolis, France, before joining ESA in Rome, Italy, as a staff member in the innovation lab of the Earth Observation directorate. There, I closely observed the transformative role of emerging technologies in the space sector, and it was in this context that quantum computing first came to my attention. This encounter ultimately led me to the joint PhD program between CERN and ESA on quantum computing, prompting me to shift my career trajectory toward research.

In 2022, I started my doctoral studies at CERN in the Quantum Technology Initiative (QTI) group led by S. Vallecorsa and M. Grossi, under the academic supervision of V. Dunjko and J. Tura in the Applied Quantum Algorithms group at Leiden University. My research developed along three main lines, two of which are at the core of this thesis. The first and primary line concerns the theoretical aspects of quantum machine learning, with a focus on the expressivity of parameterized quantum circuits and the study of quantum learning advantages. The second addresses quantum algorithms for the simulation of classical systems and related questions in complexity theory. The third explores quantum-inspired approaches, where I developed a tensor-network method for computing Betti numbers.

During my PhD, I supervised Bachelor and Master students and contributed to the teaching of introductory lectures on quantum computing. In 2024, I had the opportunity to attend the Los Alamos summer school, where I expanded my research horizon by studying the complexity of simulating interferometers with a large number of bosonic modes and connected with an international community of researchers.

In the upcoming period, I will continue my journey in quantum computing as a researcher at PsiQuantum, working on quantum algorithms for differential equations.