Introducing tools to quantify the performance of quantum computing algorithms and their applications.

PhD candidate

Katerina Gratsea

Co-supervisors

Prof. Dr. Maciej Lewenstein
Dr. Patrick Huembeli

Abstract

English

In this thesis, I focused on introducing tools to quantify the performance of quantum computing algorithms and their applications. The main focus is on two of the most popular application areas of quantum computing, quantum machine learning and quantum chemistry. To this end, I analyze the properties of quantum machine learning models by following statistical method techniques, which can help us build our understanding of the capabilities of such quantum models. Moreover, I introduce the teacher-student scheme as a computational tool to benchmark the performance of different quantum models and their training capabilities. Until large scale benchmarking is available, these tools can help us understand the potential of quantum machine learning and guide the research in the right direction. Next, in recent years substantial effort have been devoted in the develpoment of quantum algorithms for quantum chemistry applications. I introduce tools to assess the utility of various combinations of quantum chemistry algorithms. I perform extensive numerical simulations on computationally affordable systems of intermediate size to explore how quantum methods can accelerate tasks of quantum chemistry. These works set a foundation from which to further explore the requirements to achieve quantum advantage in quantum chemistry. Finally, I discuss how research in quantum computing has tended to fall into one of two camps: near-term intermediate scale quantum (NISQ) and fault-tolerant quantum computing (FTQC). Through a quantum chemistry application, I explore how to use quantum computers in transition between these two eras, namely the early fault-tolerant quantum computing (EFTQC) regime.

Español

En esta tesis, me enfoqué en introducir herramientas para cuantificar el rendimiento de los algoritmos de computación cuántica y sus aplicaciones. El principal enfoque está en dos de las áreas de aplicación más populares de la computación cuántica, el aprendizaje automático cuántico y la química cuántica. Con este fin, analizo las propiedades de los modelos de aprendizaje automático cuántico siguiendo técnicas de métodos estadísticos, que pueden ayudarnos a construir nuestra comprensión de las capacidades de dichos modelos cuánticos. Además, introduzco el esquema maestro-alumno como una herramienta computacional para evaluar el rendimiento de diferentes modelos cuánticos y sus capacidades de entrenamiento. Hasta que esté disponible el benchmarking a gran escala, estas herramientas pueden ayudarnos a entender el potencial del aprendizaje automático cuántico y guiar la investigación en la dirección correcta. A continuación, en los últimos años se ha dedicado un esfuerzo sustancial al desarrollo de algoritmos cuánticos para aplicaciones en química cuántica. Presento herramientas para evaluar la utilidad de diversas combinaciones de algoritmos de química cuántica. Realizo extensas simulaciones numéricas en sistemas de tamaño intermedio computacionalmente asequibles para explorar cómo los métodos cuánticos pueden acelerar las tareas de química cuántica. Estos trabajos establecen una base desde la cual explorar más a fondo los requisitos para lograr una ventaja cuántica en química cuántica. Finalmente, discuto cómo la investigación en computación cuántica tiende a dividirse en uno de dos campos: la computación cuántica a escala intermedia a corto plazo (NISQ) y la computación cuántica tolerante a fallos (FTQC). A través de una aplicación de química cuántica, exploro cómo usar computadoras cuánticas en la transición entre estas dos eras, específicamente el régimen temprano de computación cuántica tolerante a fallos (EFTQC).

List of Publications

1. **A. Gratsea**, V. Kasper, M. Lewenstein. Storage properties of a quantum perceptron, *arXiv:2111.08414*, (2021), accepted in PRE.

2. M. Lewenstein, **A. Gratsea**, A. Riera-Campeny, A. Aloy, V. Kasper, and A. Sanpera. Storage capacity and learning capability of quantum neural networks. *Quantum Science and Technology* **6**, (4), 045002 (2021).

3. **A. Gratsea** and P. Huembeli. Exploring quantum perceptron and quantum neural network structures with a teacher-student scheme. *Quantum Machine Intelligence* **4** (1), 1-15 (2022).

4. **A. Gratsea** and P. Huembeli. The effect of the processing and measurement operators on the expressive power of quantum models, *Quantum Machine Intelligence*, **5**, 2, 2524-4906 (2023).

5. **K. Gratsea**, C. Sun, P. D. Johnson. Evaluating the efficiency of ground state preparation algorithms, *arXiv:2212.09492*, (2022), accepted in PRA.

6. A. Katabarwa, **K.Gratsea**, A. Caesura, P. Johnson. Early-fault tolerant quantum computing, *arXiv:2311.14814*, (2023).

7. **K. Gratsea**, J. Kottman, P. Johnson, A. Kunitsa. Comparing Classical and Quantum Ground State Preparation Heuristics, *arXiv:2401.05306*, (2024).

Abstracts of the aforementioned works can be found in App. A.1.

Acknowledgments

With the rapid progress of science nowadays, the PhD dissertation on a topic is only the beggining of a long exciting journey in learning. The past 4 and half years have given me the chance to grow both personally and professionally.

First of all, I am grateful to being awarded the Marie Skłodowska-Curie fellowship, which gave me the freedom to pursue my own research directions and collaborations. And ofcourse, I would like to thank my supervisor, Maciej Lewenstein, who encouraged me to pursue these research paths and formulate my research collaborations outside of ICFO. To this end, I would like to thank all my collaborators for their time and help on the researcher projects. Namely, I would like to thank my co-supervisor Patrick Huembeli who was always responsive and approachable for any questions I had. Next, I would like to thank Peter D. Johnson, who gave me the opportunity of interning at Zapata AI and being an awesome team leader. This also opened up interesting collaborations and research projects with other researchers and Zapatistas, which I would like to thank personally: Athena Ceasura, Amara Katabarwa, Jakob Kottman and especially, Alex Kunitsa for many fruitful discussions on quantum chemistry and more.

And of course, I would like to thank a lot my family and friends for their support during these past years. Especially, my partner Akis Linardos, who has always been supportive of my endeavors and my childhood friend Maria Kanaki for being there when needed most. And ofcourse, my sister Aria Gratsea and my parents who have always been encouraging. I am also grateful to my good friends Tonia Lazopoulou and Gaby Boada. This journey would have not been as successfully and joyful as it was without the companionship of the many horses I spend time with. And mainly, my best horse-friend, Danai, and the many hours spent riding on the beach. Finally, I am grateful to my Zen masters -all of them cats. And especially my two kitties, Nyx and Yuki, for the countless hours of lap-cat support.

Table of Contents

Chapter 1

Introduction

1.1 Motivation

In recent years, quantum computing have attracted the attention of many researchers in different fields. Interestingly, a wide range of applications have been considered for quantum computing in the field of math (i.e. differential equations and prime factorization) , optimization (i.e. runtime speed-ups and enhancement of existing algorithms), machine learning (i.e. acceleration of machine learning and applications of big data problems) and chemistry (i.e. ground state approximations and estimation of systems' properties) among others.

But even though quantum computing promises to revolutionize all the aforementioned fields, it is yet unclear on which specific applications quantum computing has something to offer and what exactly it will be able to deliver. To this end, extensive research is performed to evaluate the capabilities of quantum computing in all the aforementioned fields and the required quantum computation power that will be necessary towards this end. This computation power of the quantum hardware is usually referred to as Noisy-Intermediate quantum computing (NISQ) or fault-tolerant quantum computing (FTQC) devices. NISQ corresponds to the quantum computing devices available today with approximately 100 qubits and 100 circuit depth, while FTQC refers to quantum computing devices with full quantum error correction capabilities. But what seems necessary for further advancing the field of quantum computation in a timely manner is bridging this dichotomy between NISQ and FTQC. To this end, the regime of early fault-tolerant quantum computing comes into play that we will discuss in detail in the last chapter of this thesis.

In this thesis, we introduce tools to quantify the performance of quantum computing algorithms with the goal of exploring their utility for two popular quantum computing applications, quantum machine learning and quantum chemistry. Quantum machine learning is one of the fields that has spark a lot of excitement motivated by the question whether there would be any benefit by combining the two fields of machine learning and quantum computation. Throughout this work we will focus on transforming directly machine learning concepts to their quantum counterpart, i.e. neural networks to quantum neural networks and explore their limits and capabilities. Next, we focus on quantum chemistry applications which is one of the most promising fields to showcase a potential quantum advantage, and specifically, the simulation of quantum systems which was mainly the motivation for the birth of quantum computers discussed by Feynman [1982a]. Here we perform rigorous resource estimation to quantify the feasibility of quantum algorithms and their potential quantum advantage.

This work lays the foundations of how to assess the utility of quantum computing for applications with potential practical interest.

1.2 Quantum Advantage

The practical interest of quantum computation lays on showcasing quantum advantage. Everybody wants quantum advantage. Everybody is looking for it. But the first thing to ask is:

- What do we mean by *quantum advantage*?

Interestingly, different researchers have a different idea of what quantum advantage is. Examples are:

- Run-time, i.e. showcasing a speed-up Arute *et al.* [2019b]; Goings *et al.* [2022a].
- Complexity, i.e. exploring the relation between complexity classes PP (Probabilistic Polynomial Time) and BQP (Bounded-Error Quantum Polynomial Time) Aaronson and Arkhipov [2013]; Fenner *et al.* [2009].
- Practical, i.e. perform better in a real-world task Hibat-Allah *et al.* [2023]; Amsler *et al.* [2023].
- Capacity, i.e. model's ability to express different relationships between variables Abbas *et al.* [2021]; Wright and McMahon [2020].
- Training capabilities, i.e model's ability to learn through training Pérez-Salinas *et al.* [2020]; Schuld *et al.* [2020b].
- Learning capabilities , i.e. comparing generalization performance Gili *et al.* [2023]; Benedetti *et al.* [2019b].
- Expressivity, i.e. model's ability to represent the solution Schuld *et al.* [2020b]; Sim *et al.* [2019a].

To better quantify whether we have a quantum advantage, we need to introduce tools that fairly compare the performance of quantum and classical machine learning models. The area of quantum advantage is still an emerging research field with a lot of researchers from both academia and industry actively contributing.

1.3 Classical Perceptrons and Neural Networks

Here we revisit the notion of classical perceptrons and neural networks before we delve into their quantum counterparts.

1.3.1 Classical Perceptron or Neuron

The term perceptron was first introduced by Frank Rosenblatt in 1958 Rosenblatt [1958] as a probabilistic model for information storage and organization of the brain. Interestingly, this was a side project of his main work as a psychologistRosenblatt [1957].

For the definition of the perceptron model, F. Rosenblatt took inspiration from the organization of the retina system, where the inputs come from the eye, then the processing happens and finally we get the output (or responses). Then, for the definition of the perceptron he used the same organization. This is also why he uses the term perceptron and photo-perceptron interchangeably throughout his work. Another main contribution of his is that he described information processing with probabilistic models, instead of Boolean functions which was prevailing at the time.

The more formal definition of perceptron as we use it today was first introduced in the work of Minsky and Papert in their book Minsky and Papert [1972]. Interestingly, they devoted the book to the memory of Frank's perceptron. Formally, a classical perceptron is a function that maps a N dimensional input $\vec{i}^{\mu} = (i_1^{\mu}, \ldots, i_N^{\mu})^T$ onto an output σ^{μ}, where the weight vector $\vec{w} = (w_1, \ldots, w_N)^T$ determines the information processing. The additional label $\mu \in \{1, 2, ..., p\}$ denotes different pairs of input vectors and outputs Nishimori [2001]; Minsky and Papert [1969]. They used the following activation function

$$\sigma^{\mu} = \theta\left(\vec{i}^{\mu} \cdot \vec{w} - \kappa\right) , \tag{1.1}$$

where κ is the threshold and $\theta(\cdot)$ is the Heaviside function realizing the non-linearity of the perceptron model, see Fig. 2.1a. But more activation function could be used Werbos [1974]. For example, Hopfield in his work for associative memories Hopfield [1984] introduced for the first time [Schuld] the sigmoid function

$$x_i = \mathrm{sgm}\left(\sum_{j=1}^{N} w_{ji} x_j; \kappa\right), \tag{1.2}$$

where neurons are allowed to take values in a continuous range $x_i \in [-1, 1]$.

Generally, the goal of the perceptron is to adjust the weights to perform the task at hand. To this end, different learning rules for the weights have been introduced Minsky and Papert [1988]. But different optimization algorithms could also be used to adjust the value of the weights LeCun *et al.* [1998]; LeCun [1998]. Finally, the perceptrons are simple building blocks that can be cast together to build neural network (NN) architectures. Today, this is the main way that perceptrons (or as they more often called neurons/nodes) are used. This is the content of the next chapter.

1.3.2 Neural Networks (NN)

NN are an interconnected group of perceptrons/neurons. A plethora of neural network structure have been proposed in the literature Deng and Yu [2014]. For example, popular NN structures are the Hopfield Neural Networks (HNNs) introduced in the works of Hopfield [1982]; Hopfield and Tank [1986] for information storage and retrieval. Another popular example are the feed-forward Neural Networks (ffNN) greatly used for classification tasks (Schmidhuber [2015]). Here we briefly elaborate on how the aforementioned NNs work.

Hopfield Neural Network

HNN is an interconnection of McCulloch-Pitss neurons in which the connectivity architecture obeys $w_{ij} = w_{ji}$ and $w_{ii} = 0$. This type of NN can store and retrieve information in terms of networks states. To better understand how HNN works, we can introduce the energy function of a state (x_1,...,x_N) with threshold vector (θ_1,...,θ_N) and synaptic connections $w_{ij}, i, j = 1, ..., N$ which reads

$$E\left(x_1, \ldots, x_N\right) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}x_i x_j + \sum_{i=1}^{N} \theta_i x_i. \tag{1.3}$$

Each update of the network's connections or weights w_{ij} will ideally minimize the network state or otherwise maintain it. To store a set of patterns $X^P = (x(1)^1, ..., x(1)^N), ..., (x(P)^1, ..., x(P)^N)$, the weights can be adjusted according to Hebb's learning rule Sompolinsky [1987]

$$w_{ij} = \frac{1}{P}\sum_{p=1}^{P} x_i^{(p)} x_j^{(p)}. \tag{1.4}$$

This suggests that neurons that have the same state or a high antiparallel correlation in the majority of memory patterns will receive a synaptic weight close to 1 or −1, respectively. Eventually, the Hopfield network's dynamics will converge to the minimum or ground state of the energy function, i.e. the attractor. The property $w_{ii} = 0$ makes sure that all attractors are stable states (Rojas [1996]).

As discussed in the case of perceptrons, the step function (eq. (2.1)) could be replaced by a sigmoid function (eq. (1.2)). Consequently, the activation function of a neuron x_i due to the input of neurons x_j, $j \in \{1, ..., N\}$ becomes

$$\mathrm{sgm}(a;\kappa) = \frac{1}{1 + \mathrm{e}^{-\kappa a}}, \tag{1.5}$$

which in the limit case when $\kappa \to \infty$ retrieves the step-function.

Hopfield showed that both activation models have attractor properties.

Feed-forward Neural Networks

Feed-forward Neural Networks (ffNN) are one of the first and simplest type of artificial neural networks conceived (Schmidhuber [2015]). They are formed by stacking neurons together and creating a set of input, hidden and output neurons. The name feed-forward comes from the fact that information only flows in a single direction, from the input to the output neurons. In ffNNs, each layer of neurons performs the operation $y = f(W\vec{x} + \vec{b})$ on its input $\vec{x}$, where W is the weight matrix, $\vec{b}$ is the bias and $f(\cdot)$ is the activation function of the layer.

1.3.3 Classical hidden units: a review

Deep NNs with many hidden layers of neurons have been successfully applied to different areas such as computer vision Ciregan *et al.* [2012] and reinforcement learning (Vinyals *et al.* [2019]). The reason why specifically deep neural networks perform so well on many different tasks is still an active research field in ML (Poggio *et al.* [2017]). This is a subject that we will further explore in the project related to pruning in QNN.

Classical hidden units play a crucial role in the performance and success of neural networks. Here, we focus our discussion on the notion of hidden units in Boltzmann machines (BMs) and (deep) feed-forward neural networks (ffNNs), because these two models provide a somewhat different perspective on hidden units.

BMs aim to learn the data distribution $p(\boldsymbol{x})$ or $p(\boldsymbol{x}, y)$ for supervised tasks. For a given probability distribution $p(\boldsymbol{x})$ of binary strings $\boldsymbol{x}$ (Le Roux and Bengio [2008]) there exists a parameterization of the BM to approximate $p(\boldsymbol{x})$ up to a given error ε Younes [1996]; Gu *et al.* [2020]. ffNNs on the other side are deterministic maps that approximate a function $f : X \rightarrow Y$. They are universal function approximators that can learn any continuous function on a compact space up to an error ε with only one hidden layer if it contains enough hidden neurons (Cybenko [1989]).

BMs and ffNNs highly depend on the number of hidden units for their performance. In both networks, the hidden nodes are feature extractors that capture correlations between the visible nodes. But while in ffNN the information is propagated through a layer-wise structure of hidden nodes (Younes [1996]), BMs give an extra perspective on hidden nodes as an extension of the sampling space to store conflicting patterns, known from the XOR problem in Hopfield networks (Wittek [2014]).

Since ffNNs only need one layer of hidden units to work as universal approximators the question arises why is it even necessary to build deep architectures that have more than one hidden layer. First, it is worth noting that deep architectures can always be mapped to shallow architectures and, therefore, have universal approximation capabilities just like their shallow version (**?**). In the work of Poggio *et al.* [2017], the authors study why deep architectures can learn certain tasks more efficiently than shallow NNs. They show that many hidden layers reduce the number of neurons needed at each layer and that the depth of the NN depends on the structure of the function that has to be learned. For example, the authors show how depth influences the learning of hierarchical functions of the form

$$\begin{aligned} f(x_1, x_2, \ldots, x_8) = \\ & h_3\Bigg[h_{2,1}\Bigg(h_{1,1}(x_1, x_2), h_{1,2}(x_3, x_4)\Bigg), \\ & h_{2,2}\Bigg(h_{1,3}(x_5, x_6), h_{1,4}(x_7, x_8)\Bigg)\Bigg], \end{aligned} \tag{1.6}$$

where each function $h_{i,j}$ is m times differentiable with smoothness m. They show that if one dedicates a hidden layer to each hierarchy stage of such a function, the number of neurons in the ffNN reduces from $\mathcal{O}(\varepsilon^{-n/m})$ to $\mathcal{O}((n-1)\varepsilon^{-2/m})$, where n is the number of neurons and ε is the approximation error of the function. Therefore, the depth of a NN can reduce the number of neurons that are required substantially for certain learning tasks.

In summary, we can say that the terminology "hidden node" is somewhat ambiguous. On one hand, the concept of hidden nodes in ffNNs and BMs captures the correlations of previous layers. They find fine-grained features and connect them to more coarse-grained features. On the other hand, the hidden node in BMs work as an extension of the sampling space. Both ways though implement the non-linearities needed in a NN. But while the idea of adding more hidden layers to build a deep NN does not have a straight-forward implementation on near-term quantum devices, the idea of extending the sampling space has a clear quantum analog, which is used in the Born machine (Liu and Wang [2018a]) or in dissipative QNNs (Sharma *et al.* [2020]).

1.4 Quantum Perceptrons and Neural Networks

1.4.1 Quantum Perceptrons (QPs)

The task of finding an equivalent of a perceptron unit is still an active research field. These units are mostly refereed to as quantum neurons or perceptrons and are the building blocks of quantum neural networks. An initial attempt to define a perceptron model was done by Lewenstein [1994]. The author M. Lewenstein formulated a statistical theory of QPs by processing input states into output states through unitary transforms. His work was inspired by the quantum unitary computers introduced by Lloyd [1993]. In the aforementioned work S. Lloyd introduced the necessary conditions that must be met by any computational system:

- Preparing the input states of the computer, without previous knowledge of the results of the computation.
- Transforming the input to the output states by following the internal 'computational' dynamics of the system and its interactions with its environment.
- Performing measurements on the system to extract the results of the computation.

Contrary to the classical perceptron where each physical spin encodes a classical bit, different encodings could be used for the quantum perceptron Schuld *et al.* [2015a]. For example, the straightforward realization is following the standard mapping where each classical spin is promoted to quantum spin-1/2 Pauli operator Torta *et al.* [2021]. Another approach is the binary encoding where the binary string values of the classical spins (i.e. 101) corresponds to the computational basis states (i.e.$|101\rangle$) of the wavefunction of the quantum system Wiebe *et al.* [2016]. Also, the string values of the classical spins could be mapped to the amplitudes of the computational basis states of the wavefunction which is refereed to as amplitude encoding Tacchino *et al.* [2019].

Next, one main difficulty when constructing QP models is the realization of the non-linear dynamics of the activation function while maintaining the unitary dynamics that govern the QP and also having a direct quantum hardware implementation Schuld *et al.* [2014]. In the recent years, more works have focused on the task of finding a good QP model with a direct hardware implementation. Here we mention few examples of such attempts. In the work of Yan *et al.* [2020] they introduce QPs as quantum circuits that approximate nonlinear functions, and then, propose a generalized framework to implement any nonlinear QP. In the work of Cao *et al.* [2017], the authors realized a QP by leveraging the repeat-until-success techniques for quantum gate synthesis. Another example is the recent work of Torrontegui and García-Ripoll [2019] that realizes a QP with a sigmoid activation function as an efficient, reversible many-body unitary operation which is implemented as a quasi-adiabatic passage with an Ising model. In the following chapters, we will mainly be involved with QP introduced by Tacchino *et al.* [2019] which has a direct implementation on a parameterized quantum circuit.

Exactly like their classical counterparts, QP are simple learning machines that could be stack together to form quantum neural networks. In the next chapter, we further discuss on this.

1.4.2 Quantum Neural Networks

In recent years, a lot of attention has been focused on the development of QNN. This field started to take off after the work of Kak [1995] who introduced some first ideas on the connection between neural networks and quantum mechanics. After this work was published, an increasing number of works started using the term "Quantum Neural Network" Altaisky [2001]; Andrecut and Ali [2003]; Schuld *et al.* [2014]; Chen *et al.* [2018]; Wright and McMahon [2020]; Beer *et al.* [2020]. Today the research field has expanded with a significant number of monthly contributions on QNN.

One main approach to build QNN is by combining different QP as already mentioned in the previous chapter. The recent work of Tacchino *et al.* [2020a] follows this approach to develop a feed-forward QNN. The main advantage of this approach is that there is a direct hardware realization introduced by connecting the QP. The difficulty in this case is mainly coming from realizing the non-linearity needed by following quantum mechanics in a rigorous way. In that case, QNN could also be introduced as variational quantum algorithms performing on parameterized quantum circuits. Variational quantum algorithms are quantum-classical heuristic algorithms that take advantage of both quantum and classical computation. The parameters that need to be optimized are passed on the quantum computer as parameters on the quantum circuit structure. Classical optimization techniques are used to find the optimal parameters.

This is done by optimizing a cost function that measures the difference between the output of the circuit and the desired output. Relevant works in this direction are Mitarai *et al.* [2018]; Grant *et al.* [2018]; Schuld *et al.* [2018].

But there are other approaches with a more vague connection on how classical neural networks are built. For example, quantum associative memories Ventura and Martinez [2000] are quantum algorithms that mimic the behaviour of a neural network without attempting to replicate its dynamics or structure Schuld *et al.* [2014]. The main idea in that case is to find an algorithm that could create the memory superposition

$$|M\rangle = \frac{1}{\sqrt{P}} \sum_{p=1}^{P} \left| x_1^{(p)}, \ldots, x_N^{(p)} \right\rangle \tag{1.7}$$

where

$$X^P = \left\{ \left| x_1^{(1)}, \ldots, x_N^{(1)} \right\rangle, \ldots, \left| x_1^{(P)}, \ldots, x_N^{(P)} \right\rangle \right\} \tag{1.8}$$

are the P patterns to be stored. The main advantage in that case is that these models perform as quantum realization of classical neural networks, but the bottleneck could be that the hardware realization might be difficult or expensive in terms of resources. Relevant works in this direction are Rebentrost *et al.* [2018]; Aguilar *et al.* [2020].

1.4.3 Non-linearities in QNNs

Classical hidden units play a crucial role in the performance and success of neural networks as already explained in chapter 1.3.3. For example, in Boltzmann machines and feed-forward neural networks, the hidden nodes are feature extractors that capture correlations between the visible nodes. But, while in ffNN the information is propagated through a layer-wise structure of hidden nodes Younes [1996], BMs give an extra perspective on hidden nodes. They extend the sampling space to store conflicting patterns, as happens with the XOR problem in Hopfield networks Wittek [2014]. While adding more hidden layers to build a deep NN does not have a straightforward implementation on near-term quantum devices, extending the sampling space has a clear quantum analog already used in the Born machine Liu and Wang [2018a] or in dissipative QNNs Sharma *et al.* [2020].

To build deep structures in QNNs, non-linearities will play a crucial role, as in their classical counterpart. The question that often arises is how to achieve the non-linearities needed in quantum circuits when all the operations are unitary. Just because we have a linear representation of a gate, does not necessarily imply that the gate or the function it computes is linear. A classical logical gate or circuit computing the function f by definition is linear if $f(x \oplus y) = f(x) \oplus f(y)$, where $\oplus$ is the bit wise XOR operation Patel *et al.* [2003]. One can, therefore, easily show that for example the Toffoli gate is not linear if applied on binary input strings. Apart from entangling gates, the measurement Tacchino *et al.* [2019]; Chen *et al.* [2018]; Schuld *et al.* [2020a] and the data encoding can introduce non-linearities on NISQ devices. Specifically, the data encoding can introduce non-linearities if the gates are non-linear functions of the data input x. For example, if one rotation of the data is performed, the transformation could be linear (for example, with the reflection matrix) and the relative distance of the data is not altered. Instead, if the unitaries perform at least two rotations over different rotation axes the gates are a non-linear function of the data input x Havlíček *et al.* [2019]; Pérez-Salinas *et al.* [2020, 2021a]. This ensures the non-linearity of the feature map $x \longrightarrow \psi(x)$ in the Hilbert space of the quantum system Schuld [2021a].

1.5 Gardner's program

Statistical mechanics was introduced to study classical neural networks (NN) in the 1980s, mostly focusing on Hopfield-like attractor ANN Hopfield [1982]; Amit [1989]. Then, Hopfield networks were studied as constraint satisfaction networks and also cognitive models, e.g. as a simple model for memory, but have never obtained any relevance for machine learning. Their relationship with contemporary machine learning is typically considered to be remote. Restricted Boltzmann machines are related to them and do not fall out of fashion completely (cf. Decelle and Furtlehner [2021]; Pozas-Kerstjens *et al.* [2021] and references therein), but their working only partially relies on capacity estimates for Hopfield networks.

Recently, this situation starts to change with the progress of statistical methods that can be applied to classical feed-forward NNs and deep learning models Tishby. Initially, feed forward NNs and deep learning (cf. Rumelhart *et al.* [1986]; McClelland *et al.* [1986]) were considered very separately from ANNs, but nowadays statistical physics methods are being applied to feed forward NNs (Tishby; Carleo *et al.* [2019]). ANNs, and even the simple perceptrons, are back in the centre of interests with the advent of quantum technologies, due to the possible realization of simple quantum NNs with ultracold atoms, trapped ions, Rydberg atoms, super-conducting qubits, or photonic systems (cf. Fraxanet *et al.*). Importantly, tools of statistical physics have already established deep relations between neural networks, spin glasses, complexity, and information processing (Nishimori [2001]; Müller *et al.* [1995]). One advantage of statistical physics is the computation of global properties of physical systems without knowing the microscopic details. Recently, there has been a true revival of increasing interest in using statistical physics techniques to study quantum information problems (Aubin *et al.* [2019]; Carleo *et al.* [2019]).

One important application of statistical physics to information processing concerns the Hopfield-like networks or even simpler perceptrons. For example, there are many learning rules Davey and Adams [2004] for the Hopfield network, which can be used to obtain a desired input-output relations. Still, the connection between the global properties of the perceptron, and a specific learning rule might be challenging to analyze.

In the seminal works Gardner [1988]; Gardner and Derrida [1988] Gardner addressed this challenge by using statistical physics to calculate the maximum storage capacity of a Hopfield network without referring to any specific learning rule. This approach of analyzing artificial networks without specifying the learning rule and treating the weights as a random variable is frequently referred to as Gardner's program. So far, the maximum storage capacity has also been considered for other QNN models Sompolinsky [1987]; Aubin *et al.* [2019]; Ding and Sun [2019]. The work of Gardner demonstrated that we can study Hopfield networks independent of the precise learning rule that is used. But, it is not only of considerable historical significance: it sheds light on the most important questions of contemporary quantum ML Schuld and Killoran [2022a]. Moreover, it is extremely general, adaptive and versatile; so far it has been applied for very different models of quantum perceptrons or quantum neural networks, or even to calculate volumes of quantum correlated (entangled) states. Here is a list of selected examples and applications of Gardner's program for various models of quantum perceptrons, quantum NN and more:

- In ref. Lewenstein [1994] quantum perceptron is defined as a unitary map followed by projective measurments in a multidimensional Hilbert space. Calculation of the relative volume reduces to calculation of the volume in the unitary group space.

- Gardner's relative volume approach clearly inspired the pioneering attempts to estimate volume of quantum correlated states, such as entangled states Życzkowski *et al.* [1998]. Integration consists in the first place in integration over unitary group in very high dimension, though.

- Recently, Gardner's program has been used on QNN models Lewenstein *et al.* [2021] corresponding to completely positive trace preserving maps (CPTP). Here relative volume requires integration over the space of maps.

- Gardner's program inspired investigation of the relative volume of parent Hamiltonians having a target ground state up to some fixed error ϵ Díaz *et al.* [2021].

- A careful look on other models of quantum perceptrons immediately suggests that Gardner's approach is possible and might turn out to be useful as well. For instance, for the quantum perceptron models introduced in Torrontegui and García-Ripoll [2019]; Beer *et al.* [2020] integration over the unitary group is needed to realize the Gardner's program.

- In the present paper, Gardner's program is applied to the specific quantum perceptron model proposed in Tacchino *et al.* [2019].

By studying these models, we are one step closer to understand whether they provide a hope for quantum advantage.

1.6 Quantum Algorithms

The history of algorithmic development suggests that great ingenuity is needed to come up with algorithms even for simple tasks, such as the multiplication of two numbers Schönhage and Strassen [1971]. The development of quantum algorithms comes with the extra constraint that the algorithms should run on a quantum computer and provide an advantage over their classical counterpart. Also, taking into account the nature of quantum mechanics, makes it even more difficult to construct quantum algorithms.

Even so, the development of quantum algorithms is an active research field for the past 30 years Montanaro [2016] with significant progress. Already more than 200 quantum algorithms are mentioned in the website qua [Accessed 2023]. One of the early developments of quantum algorithms were Deutsch's Deutsch [1985] for determining whether a Boolean function is constant or balanced, Grover's Grover [1996] for performing quantum searching and Shor's Shor [1997] for factoring large numbers. Here we discuss more recent implementations of quantum algorithms.

1.6.1 Variational Quantum Eigensolver (VQE)

The VQE algorithm is a quantum classical heuristic method that was initially introduced in Peruzzo *et al.* [2014a] and later extened in McClean *et al.* [2016]. The goal of the algorithm is to find an upped bound for the lowest expectation value of an observable of a given Hamiltonian $\hat{H}$ with respect to a parameterized wavefunction $|\psi\boldsymbol{\theta}\rangle$

$$E_0 \leqslant \frac{\langle\psi(\boldsymbol{\theta})|\,\hat{H}\,|\psi(\boldsymbol{\theta})\rangle}{\langle\psi(\boldsymbol{\theta})|\psi(\boldsymbol{\theta})\rangle}.. \tag{1.9}$$

Essentially this problem reduces to finding a good parametrization of the wavefunction $|\psi(\boldsymbol{\theta})\rangle$ which ideally would correspond the ground state energy of the Hamiltonian. The outline of the algorithm is:

- Prepare a quantum state $|\psi(\boldsymbol{\theta})\rangle$ on a quantum computer with an initial set of parameters θ_0.
- Measure the expectation value
- Use a classical optimizer to update the parameter values.
- Iterate this procedure until convergence with respect to the value of energy.

Then, the given parameters θ' in the converged state define the desired quantum state.

In order to implement the VQE algorithm using a quantum computer, the wavefunction $|\psi(\boldsymbol{\theta})\rangle$ is represented by a parameterized unitary operator $U(\boldsymbol{\theta})$. This unitary operator is applied to an initial state of N qubits, with $\boldsymbol{\theta}$ ranging from $[-\pi$ to $\pi]$. Typically, the initial state used is $|0\rangle^{\otimes N}$. The unitary operator U is then applied to the initial state and we can define the VQE cost function as

$$E_{\mathrm{VQE}} = \min_{\boldsymbol{\theta}} \langle \mathbf{0}| U^{\dagger}(\boldsymbol{\theta})\hat{H}U(\boldsymbol{\theta}) |\mathbf{0}\rangle \,. \tag{1.10}$$

In order for the hybrid nature of VQE to become clear, we can further extend the description of the VQE algorithm applied on a quantum computer. Specifically, we can write the Hamiltonian as a tensor product of Pauli operators, which are directly measurable on a quantum computer. The Hamiltonian can be rewritten as

$$\hat{H} = \sum_{a}^{\mathcal{P}} w_a \hat{P}_a , \tag{1.11}$$

with w_a a set of weights, and $\mathcal{P}$ the number of Pauli strings: $\hat{P}_a \in \{I, X, Y, Z\}^{\otimes N}$ in the Hamiltonian with N the number of qubits used to model the wavefunction. Then, the cost function Eq. (1.10) becomes

$$E_{\mathrm{VQE}} = \min_{\boldsymbol{\theta}} \sum_{a}^{\mathcal{P}} w_a \langle \mathbf{0}| U^{\dagger}(\boldsymbol{\theta})\hat{P}_a U(\boldsymbol{\theta}) |\mathbf{0}\rangle \,. \tag{1.12}$$

Then the expectation value of a Pauli string $\hat{P}_a$ is computed directly on a quantum computer, while the summation and minimization $E_{VQE} = \min_{\boldsymbol{\theta}} \sum_a w_a E_{P_a}$ is computed on a conventional computer. This makes evident the hybrid nature of the VQE algorithm Tilly *et al.* [2022].

Further research on VQE algorithms have identified certain issues that are challenging to overcome:

- The choice of circuit ansatz:
  The first challenge is the choice of the circuit ansatz. It is important to find parameterized quantum circuits that have the expressive power to yield a sufficiently good approximation to the optimal solution and keep the number of entangling gates and circuit depth as low as possible. Some works have proposed different quantum ansatzes that are problem and/or hardware-inspired Sim *et al.* [2019b]. Other recent works focus on systematically optimizing the circuit architecture. But, in most cases, this results in a significant overhead in the classical optimization runtime Grimsley *et al.* [2019].

- Classical optimization runtime:
  As already mentioned above, the classical optimization runtime is another issue of VQA. It was recently shown that VQA optimization is inherently NP-hard, which means that VQA optimization is intrinsically difficult and does not merely inherit the hardness of the given problem Bittel and Kliesch [2021]. This, though, does not prove that VQA optimization is beyond hope- specific solutions have been found for other NP-hard problems. This results emphasizes that VQA optimization is a difficult problem to handle.

- Barrean plateaus:
  Another problem related to the training of VQA is that partial derivatives of an objective function based on a random parameterized quantum circuit have very small valuesMcClean *et al.* [2016]. Therefore, their training landscape includes large regions where the gradient is almost zero, but which do not correspond to a global or local minimum. This problem makes them difficult to train with gradient-based or related local optimization methods. Also, this exponential decrease in the variance of gradients hinders the extraction of meaningful values from quantum hardware, especially in the case of current hardware that is subject to noise.

- The measurement problem:
  Finally, another potential bottleneck of VQE is the measurement problem Gonthier *et al.* [2020]. Recent works show that a huge number of measurements is needed to provide good energy estimations. This in turn needs large runtime which is well-beyond the coherent time of current quantum devices, and most probably of the runtime that NISQ devices are hoped to work for.

Separable Pair Approximation (SPA) algortihm

The recent work of Kottmann and Aspuru-Guzik [2022a] which gives a variation of the VQE algorithm optimized using a separable pair approximation (SPA). Then, once combined with specific circuit compilation strategies can give classically tractable circuit classes with very short circuit depths. Here we discuss the main elements of the SPA algorithm.

Let's assume that we have a system with N electrons. Then we can take the pair of $N/2$ electrons and write the separable pair (SP) wavefunction as a tensor product of these pairs:

$$|\Psi_{\text{SP}}\rangle = \bigotimes_{k=1}^{N_{e/2}} |\psi_k\rangle \tag{1.13}$$

where $|\psi_k\rangle$ are electron pair functions. Next each pair can be represented as a summation of spin-orbitals, which essentially are one-electron functions

$$|\psi_k\rangle = \sum_{mn} c^k_{mn} |\phi^k_m\rangle \otimes |\phi^k_n\rangle \tag{1.14}$$

with the extra assumption that all orbitals are orthonormal

$$\langle\phi^k_l\rangle\, \phi^{k'}_{l'} = \delta_{kk'}\delta_{ll'}. \tag{1.15}$$

Interestingly, while the wavefunction has a product structure, the individual pair-functions are coupled through the Hamiltonian. Finally, to realize this on a quantum device we need to find the unitaries U_k that create the pair-functions $|\psi_k\rangle$ starting from the initial state $\psi_{init} = HF\,|00\ldots0\rangle$. Therefore, we have

$$|\Psi_{\mathrm{SP}}\rangle = \prod_{k}^{N_{e/2}} U_k\left(\boldsymbol{\theta}_k\right) U_{\mathrm{HF}}\,|00\ldots0\rangle\,. \tag{1.16}$$

The unitaries U_k can be realized on quantum hardware through one- and two-electron excitation gates. An elementary n-electron excitation gate is given by

$$U_{\boldsymbol{pq}}(\theta) = e^{-i\frac{\theta}{2}G_{\boldsymbol{pq}}} \tag{1.17}$$

which essentially describes excitations between spin orbitals $\boldsymbol{p} = \{p_0, p_1, \ldots, p_n\}$ and $\boldsymbol{q} = \{q_0, q_1, \ldots, q_n\}$ with $p_k \neq q_l \forall (k, l)$ through the fermionic excitation generator

$$G_{\boldsymbol{pq}} = i\left(\prod_{k} a^{\dagger}_{p_k} a_{q_k} - \text{ h.c. }\right). \tag{1.18}$$

Now we have all the elements needed to describe the SPA algorithm. Finally, the VQE cost function can be expressed as

$$E = \min_{\boldsymbol{\theta}} \langle \Psi_{\mathrm{SP}}(\boldsymbol{\theta}) | H | \Psi_{\mathrm{SP}}(\boldsymbol{\theta}) \rangle\,, \tag{1.19}$$

which minimizes the expectation value of the parameterized product of pair-functions over the full electronic Hamiltonian.

1.6.2 Low-depth boosters

Apart from VQE-type algorithms there other algorithm for state preparation. In this section, I will briefly elaborate how the low-depth booster Wang *et al.* [2022b] works as a quantum state preparation method. The main idea of this method is to compose a circuit that implement a function of H so that it boosts the overlap with the ground state of H, while suppresses the rest of the eigenstates. This approach is what the authors refer to as boosters.

The aforementioned problem can be introduced as follow. Assume a Hamiltonian $H = \sum_{j=1}^{D} \lambda_j \left|\lambda_j\right\rangle \left\langle\lambda_j\right|$ with eigenvalues λ_j's and eigenstates $\left|\lambda_j\right\rangle$'s, where $a \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_D \leq b$ for some $a, b \in \mathbb{R}$. Then let $\left|\psi\right\rangle = \sum_{j=1}^{D} \mu_j \left|\lambda_j\right\rangle$ be the state given by an ansatz circuit such that $\mu_1 \neq 0$.

The goal is to find a function $f : \mathbb{R} \to \mathbb{C}$ such that $|f|$ decreases monotonically on the interval $[a, b]$. Then implementing the operation $f(H)$ on the state $\left|\psi\right\rangle$ yields the unnormalized state

$$f(H)\left|\psi\right\rangle = \sum_{j=1}^{D} \mu_j f(\lambda_j)\left|\lambda_j\right\rangle, \tag{1.20}$$

whose normalized version is

$$\frac{f(H)\left|\psi\right\rangle}{\|f(H)|\psi\rangle\|} = \sum_{j=1}^{D} \mu_j' \left|\lambda_j\right\rangle = \frac{1}{\sqrt{Z}} \sum_{j=1}^{D} \mu_j f(\lambda_j)\left|\lambda_j\right\rangle, \tag{1.21}$$

where $Z = \sum_{j=1}^{D} |\mu_j f(\lambda_j)|^2$ and $\mu_j' = \mu_j f(\lambda_j)/\sqrt{Z}$.

Then, the amplitudes of the lower-energy eigenstates of H in $\frac{f(H)|\psi\rangle}{\|f(H)|\psi\rangle\|}$ are larger than their counterparts in $\left|\psi\right\rangle$. In particular, the overlap between $\frac{f(H)|\psi\rangle}{\|f(H)|\psi\rangle\|}$ and the ground state of H is larger than the one for $\left|\psi\right\rangle$. This can be seen by following the assumptions on f gives $\left|\frac{\mu_i f(\lambda_i)}{\mu_j f(\lambda_j)}\right| \geq \left|\frac{\mu_i}{\mu_j}\right|, \forall i \leq j$. In turn it follows that $\|\mu_j'\| \geq \|\mu_j\|$ for $j = 1, 2, \ldots, k$, for some $1 \leq k \leq D$.

While any function f that satisfies the monotonicity condition can theoretically be used to boost a Hamiltonian, the practical cost of implementing the operation $f(H)$ can vary widely depending on the specific f chosen. Additionally, $f(H)$ is generally non-unitary, meaning that it cannot be implemented with certainty. Therefore, a careful balance the following factors is necessary when choosing the function f:

- the overlap increment with the ground state (or the low-energy eigenstates) of H that $f(H)$ can achieve
- the circuit depth implementing $f(H)$
- the success probability implementing $f(H)$

In the same work Wang *et al.* [2022b], the authors discuss how to implement the function f once it is chose. Given a function f, the Fourier approximation can be written as

$$f(x) \approx \sum_{j=1}^{K} \alpha_j e^{it_j x} \tag{1.22}$$

for some $\alpha_j \in \mathbb{C} \backslash \{0\}$ and $t_j \in \mathbb{R}$. Then we can also have a Fourier approximation of the booster operation $f(H)$:

$$f(H) \approx \sum_{j=1}^{K} \alpha_j e^{it_j H}. \tag{1.23}$$

Therefore $f(H)$ is expressed as a linear combination of the unitary operations $e^{it_j H}$'s. This suggests that the LCU method can be used to implement $f(H)$ approximately and probabilistically.

Finally, in Chapter 6 we specifically use the Gaussian function and its Fourier transform:

$$f_a(x) = e^{-ax^2} \leftrightarrow \hat{f}_a(\xi) = \sqrt{\frac{\pi}{a}} e^{-\frac{(\pi\xi)^2}{a}}; \tag{1.24}$$

as the booster function f and a parameterized parameter α. Note that technically the "truncated Gaussian boosters" are non-monotonic (due to the truncation), but as the authors emphasize non-monotonic boosters might ensure a good performance to cost ratio if they can be realized with short-depth quantum circuits. In the case of "truncated Gaussian boosters" their corresponding functions have steeply decreasing envelopes, ensuring their performance is not compromised.

Once we find the optimal parameter a, the corresponding booster operation $f_a(H)$ is implemented by the linear combination of unitaries (LCU) method based on the equation:

$$f_a(x) = \sqrt{\frac{\pi}{a}} \int_{-\infty}^{\infty} e^{-\frac{(\pi\xi)^2}{a}} e^{2\pi i x \xi} d\xi \approx f_{T,N;a}(x) = \frac{T}{N}\sqrt{\frac{\pi}{a}} \sum_{j=-N}^{N-1} e^{-\frac{(\pi\xi_j)^2}{a}} e^{2\pi i x \xi_j}, \tag{1.25}$$

where $\xi_j = (j+1/2)T/N$, and N is sufficiently large so that $f_a(x) \approx f_{T;a}(x) \approx f_{T,N;a}(x)$ for all $x \in [0,1]$.

Interestingly, the circuit operation implemented by the LCU method used in the Gaussian booster has the same structure as the phase estimation circuit used in Quantum Phase estimation (QPE) algorithm discussed in the next section.

1.6.3 Quantum phase estimation algorithm

Quantum phase estimation (QPE) algorithms, or more generally energy estimation algorithms, solve a problem that is interesting both from a physical point of view, but also with significant industrial relevance. For example, energy estimation is useful for the simulation of molecules and materials Cao *et al.* [2020]. Also, other interesting problems can be reduced to the task of QPE, such as factoring Nielsen and Chuang [2011]. What is the task of QPE?

The aforementioned algorithms can estimate the eigenvalue associated to a given eigenvector of a unitary operator. Formally, the goal of QPE algorithm is given an eigenvector ψ of a unitary operator U to find the eigenvalue $e^{2\pi i\theta}$ of $|\psi\rangle$, i.e $e^{2\pi i\theta} = \langle\psi|U|\psi\rangle$.

In most cases thought we do not have the exact ground state of the system, but rather an approximation. Therefore, we need a good trial initial state $|\psi_i\rangle$ that has significant overlap with the true ground state of the studied Hamiltonian. The QPE algorithm consists of a set of qubits that are necessary to encode the trial input state along with a set of control qubits (usually referred to as ancilla qubits) initiated to $|0\rangle$. The initial state of the system is $|0\rangle^{\otimes n}|\psi\rangle$.

The first step of the algorithm is to apply the Hadamard gate to each qubit of the control qubits. The state of the system becomes

$$\frac{1}{2^{\frac{n}{2}}}(|0\rangle + |1\rangle)^{\otimes n}|\psi_i\rangle = \frac{1}{2^{n/2}}\sum_{j=0}^{2^n-1}|j\rangle|\psi_i\rangle. \tag{1.26}$$

Then we perform a set of controlled time evolutions of the studied Hamiltonian H on the trial state. This could be interpreted as a black box which performs a controlled- U^j operation for an integer j

$$U_j = e^{-2^{(j-1)}i\tau H}. \tag{1.27}$$

This black box operation results to

$$|\phi\rangle = \frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}e^{2\pi ij\theta}|j\rangle|\psi_i\rangle \tag{1.28}$$

Finally, we apply the inverse Fourier transform and perform a measurement in the computational basis on the controlled qubits. The output of the aforementioned procedure is a n-bit approximation $\widehat{\theta}$ to θ.

One significant concern of the aforementioned procedure is the runtime necessary to perform the algorithm. This scales as $\tilde{O}\left(\epsilon^{-1}\gamma^{-2}\right)$, where ϵ is the target accuracy and γ is the overlap between the trial input quantum state with the true ground state of the Hamiltonian. The dependence ϵ^{-1} is referred to as the Heisenberg-limited precision scaling which is the optimal that could be realized. Variations of the QPE algorithm along with other algorithms for ground state energy estimation can be found in Gilyén *et al.* [2015] and references therein.

Chapter 2

Storage capacity of QP and QNNs

In recent years there has been an increasing interest in the field of Quantum Machine Learning (QML) which either refers to classical ML techniques applied to quantum physics problems or ML concepts implemented directly on quantum hardware. Here we are involved with the latter and we are interested in exploring whether there could be any potential quantum advantage emerging from such implementations.

Specifically, we transform the building block of classical neural networks, the perceptron, to its quantum version, i.e. the quantum percepton and ask whether there could be any advantage over its classical counterpart in terms of its storage properties, i.e. the number of patterns it could store. To explore this question we revisit the techniques used to describe the properties of a classical perceptron, essentially by applying statistical physics techniques. Therefore, we follow the same approach and apply statistical physics techniques to this quantum information problem. Finally we explore the storage properties of quantum neural networks.

In this chapter we introduce techniques that could bridge the gab between quantum computation and learning theory.

2.1 Introduction

As the perceptron is the simplest learning machine, it is of great value to explore its storage capacity, i.e. the ratio α of the patterns p that could be stored over the computational resources. In the classical case, the computational resources are equivalent to the number of physical spins N, and thus the patterns that could be stored are $p = \alpha N$. In the quantum case, the computational resources are equal to the number of the computational basis $m = 2^N$, and thus $p = \alpha m = \alpha 2^N$. We already see the exponential dependence on the number of physics spins N for the quantum perceptron, but a fair comparison between these two perceptrons is over the computational resources.

In Sec. 2.2, a potential quantum advantage is explored through the notion of maximal storage capacity of the quantum perceptron versus its classical counterpart. To do so, we apply statistical physics techniques to this information problem. One advantage of the statistical physics approach is that one can frequently make statements about the system's global properties without knowing the microscopic details. Specifically, we analyze the quantum perceptron without specifying the learning rule and treating this ignorance as a statistical physics problem which is the Gardner program, introduced in Sec. 1.5.

As with perceptrons, one crucial feature of NNs is their storage capacity for associative memory, that is, the number of patterns (stored memories/attractors) the network has for a given number of neurons n. Despite recent progress, the storage properties of generic QNN is still an open question. In Sec. 2.3, a potential quantum advantage is explored through the number of patterns that could be stored. We address this question by associating attractor QNNs to CPTP, where the total volume of attractor QNN corresponds to the volume of CPTP maps that have a fixed set of stationary states. We analyze the learning capability, i.e. the number of patterns that could be learned by the QNN by applying Gardner's program and estimating the relative volume of QNNs realizing the desired attractor input-output relations.

The original results presented in this chapter are based on the publication [Gratsea *et al.*, 2021] with V. Kasper and M. Lewenstein and [Lewenstein *et al.*, 2021] with M. Lewenstein, A. Riera-Campeny, A. Aloy, V. Kasper and A. Sanpera.

2.2 Gardner's program applied to a quantum perceptron architecture

In this work, we apply Gardner's program to a specific quantum analog of the classical perceptron introduced in Tacchino *et al.* [2019] to explore its storage capacity. This quantum perceptron model has a direct implementation on quantum hardware Tacchino *et al.* [2019] and uses amplitude encoding which is beneficial in terms of memory resources as mentioned earlier. But, for this perceptron model it is not clear how the storage capacity will compare to its classical counterpart contrary to the quantum perceptron models Torta *et al.* [2021]; **?**, which have the same storage capacity. Therefore, we aim to perform a reasonable comparison of the maximum storage capacity of this quantum perceptron model with its classical counterpart. This work brings us one step closer to understanding whether they provide hope for quantum advantage and sheds light on the most important questions of contemporary quantum machine learning models as general learning machines Schuld and Killoran [2022a].

2.2.1 Perceptron model

Classical perceptron model

A classical perceptron is a function that maps a N dimensional input $\vec{i}^{\mu} = (i_1^{\mu}, \ldots, i_N^{\mu})^T$ onto an output σ^{μ}, where the weight vector $\vec{w} = (w_1, \ldots, w_N)^T$ determines the information processing. The additional label $\mu \in \{1, 2, ..., p\}$ denotes different pairs of input vectors and outputs Nishimori [2001]; Minsky and Papert [1969]. Moreover, we consider the following activation function

$$\sigma^{\mu} = \theta \left(\vec{i}^{\mu} \cdot \vec{w} - \kappa \right) , \tag{2.1}$$

where κ is the threshold and $\theta(\cdot)$ is the Heaviside function realizing the non-linearity of the perceptron model, see Fig. 2.1a.

Quantum perceptron model

A quantum analog of the classical perceptron Tacchino *et al.* [2019] is depicted in Fig. 2.1b with the corresponding quantum circuit in Fig. 2.1c. In this quantum perceptron the connection between the inputs, outputs and weights is given by the activation function

$$\sigma^{\mu} = \theta \left(\tfrac{1}{m} |\vec{i}^{\mu} \cdot \vec{w}|^2 - \kappa \right) , \tag{2.2}$$

where the non-linearity of the perceptron is realized by the measurement, see Fig. 2.1b.

In the quantum case the input vector is $\vec{i}^{\mu} = (i_0^{\mu}, \ldots, i_{m-1}^{\mu})^T$ and the weight vector is $\vec{w} = (w_0, \ldots, w_{m-1})^T$, where m is the dimension of the Hilbert space. The vectors $\vec{i}^{\mu}$ and $\vec{w}$ are encoded in quantum states

$$|\psi_{\vec{i}}\rangle = \frac{1}{\sqrt{m}} \sum_{j=0}^{m-1} i_j \, |j\rangle , \tag{2.3a}$$

$$|\phi_{\vec{w}}\rangle = \frac{1}{\sqrt{m}} \sum_{j=0}^{m-1} w_j \, |j\rangle , \tag{2.3b}$$

respectively with the orthonormal basis vectors $|j\rangle$ form the computational basis and we focused on the case of binary inputs and weights. The encoding unitary $U_{\vec{i}}$ prepares the input state, while the processing unitary $V_{\vec{w}}$ computes the inner product between $\vec{i}$ and $\vec{w}$. The precise definition of the unitaries can be found in App. B.2. After the encoding and processing step (depicted by two blocks in Fig. 2.1c), a multi-controlled NOT gate is applied between the register and an ancilla qubit. Measuring the ancilla qubit in the computational basis gives $|1\rangle$ with probability $|\vec{i}^{\mu} \cdot \vec{w}|^2$, see Fig. 2.1c and App. B.2 for details.

2.2.2 Activation functions

It is important to discuss the differences between the classical activation function and the quantum (i.e., quadratic) activation function. In the classical case, the perceptron is a linear classifier, and its activation function depends linearly on the weighted signal. In general, the quadratic activation function introduces a nonlinearity to the model. We refer here to Ref. Müller *et al.* [1995], where the concept of a quadratic activation function has already been discussed.

In general, according to Ref. Müller *et al.* [1995], quadratic, or more generally nonlinear activation function might improve or deteriorate the performance of the classical perceptron. In fact, recent works have explored a quadratic activation function of a classical neuron. Even though their increased representation and efficiency Fan *et al.* [2018], they have increased computational costs and restricted expressive abilities Fan *et al.* [2020]. Regarding the storage capacity, the authors in the work of Fan *et al.* [2020] discuss that a two-layer neural network with N inputs, K hidden units, binary outputs, and a quadratic activation function results in the same value of the storage capacity with the quantum perceptron. Moreover, taking into account that $\alpha_{c,\max}$ has a finite value for the quantum perceptron and that m equals the dimension of the Hilbert space ($m = 2^N$), the number of patterns that can be stored is exponential in the number of spins N in agreement with recent works Andrecut and Ali [2003]; Ventura and Martinez [2000]; Wright and McMahon [2020].

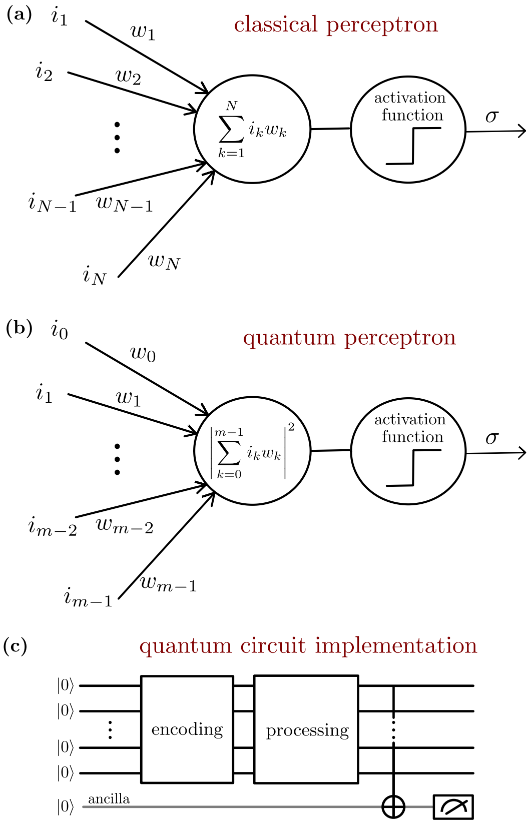


Figure 2.1 Classical and quantum perceptrons (a) Schematic outline of the classical perceptron: An N-dimensional input array $\vec{i}$ is processed with a weight vector $\vec{w}$ such that $\vec{i} \cdot \vec{w}$ enters the activation function. **(b)** Schematic outline of the quantum perceptron: An m-dimensional input array $\vec{i}$ is processed with a weight vector $\vec{w}$ to produce the inner product squared of these vectors. Both cases, the classical and the quantum, employ a non-linear activation leading to the output σ. **(c)** Quantum circuit implementation of the quantum perceptron following the work of Tacchino *et al.* [2019]. An encoding unitary realizes the input state $|\psi_{\vec{i}}\rangle$ and the processing unitary computes the inner product of the input and weight vectors. The outcome is then written on the ancilla qubit with a multi-controlled NOT gate. Finally, the activation is measured by the readout of the ancilla qubit.

For example, in the recent work Lewenstein *et al.* [2021] the authors applied Gardner's program to attractor Quantum Neural Networks and found that the learning of P patterns at the order of $2^{N/2}$ is possible.

The same quadratic activation function that we use for quantum perceptron can be directly implemented in a classical perceptron, but it does not seem to make a lot of sense. Our activation function is directly related to quantum measurement theory: it estimates how much a given set of normalized weights can be similar (in the sense of the squared scalar product) to a random pattern. This is in principle a geometrical question and be considered in a purely classical system, but it is not the true goal for introducing nonlinear activation functions.

2.2.3 Gardner's program

The correct choice of the weights results in a desired input-output relation, i.e., a specific mapping between the input $\vec{i}^{\mu}$ and the output σ^{μ}. A learning rule is usually applied to find the correct weights, such as the Hebbian rule Sompolinsky [1987]. While the Hebbian rule has an appealing simplicity, Gardner, in her works Gardner [1988]; Gardner and Derrida [1988], was interested in the global properties of the classical perceptron model without specifying the learning rule. She asked the question: What is the maximum number of input-output patterns that the classical perceptron can realize? Therefore, she considered the relative volume in the space of possible weights, which realizes a given input-output relation.

2.2.4 Storage capacity

The problem of storage capacity goes indeed back to the theory of classical perceptrons Minsky and Papert [1972]. In the "classic" paper from 1964 Cover [1965], T.M. Cover demonstrated using simple geometrical arguments that the separating capacities of families of nonlinear decision surfaces by a direct application of a theorem in classical combinatorial geometry. It is shown that a family of surfaces having N degrees of freedom has a natural separating capacity of $2N$ pattern vectors, thus extending and unifying earlier results of others on the pattern-separating capacity of hyperplanes. Thus, the critical storage capacity of a classical perceptron is $\alpha_c = 2$.

The problem of storage capacity returned in 1982 in the seminal paper of J. Hopfield Hopfield [1982], who has shown numerically that the Hopfield model with the, so-called, Hebbian learning rule may store $0.14N$ random patterns. E. Gardner came back to the problem analyzing the shrinking in the volume of perceptrons that correctly reproduce the desired input-output relations normalized to the volume of connection vectors $\vec{w}$. The advantage of the work of Gardner was that it calculated storage capacity independently of the learning rule used. It also reproduced correctly the classical geometrical bound of T.M. Cover Cover [1965].

Following Gardner's work, the storage capacity can be obtained from the fraction of $\vec{w}$-space which *correctly* and *exactly* reproduces the desired input-output relations normalized to the volume of vectors $\vec{w}$. When increasing the number of patterns, the volume of vectors $\vec{w}$ typically shrinks, and the relative volume of the weights vanishes. The limit of vanishing relative volume defines the storage capacity of the perceptron Nishimori [2001].

From the definition of the storage capacity, the difference between the classical and quantum perceptrons results from two aspects of to the definition of quantum perceptron proposed in Tacchino *et al.* [2019]. The classical perceptron checks whether the signal corresponding to a given input pattern has an appropriate sign at the output. Quantum perceptron uses quantum measurement principles, and checks for a given input pattern how big are the quantum overlaps of output states. Quantum perceptron by definition does not reproduce the input patterns *correctly* and *exactly*; it does it with certain optimal error, or better to say accuracy.

The other important aspect is that of the different dimensionality of the input vectors, which equals N physical inputs for the classical perceptron. In contrast, in the quantum case, the number of inputs equals the dimension of the Hilbert space m. Hence, for the classical perceptron we have $\alpha_c = p/N$ (i.e. the capacity p is proportional to the number of spins N), while for the quantum perceptron $\alpha_c = p/m$ (i.e. the capacity is exponential in the number of spin configurations, $p \propto m = 2^N$). For the classical perceptron, the storage capacity is known to be 2 (for $\kappa = 0$) and was calculated for example in Cover [1965]; Gardner [1988]; Gardner and Derrida [1988]; Müller *et al.* [1995]; Nishimori [2001]. More, precisely in the classical perceptron, when $\alpha > \alpha_c(\kappa)$, the relative volume shrinks abruptly to zero. In contrast, when $\alpha \leq \alpha_c(\kappa)$, the relative volume is non-zero but shrinks moderately slowly exponentially with m.

It is rather difficult to compare this situation with our results obtained for the quantum perceptron. First of all, critical capacity and the nature of the phase transition depend on the form of weights we use (spherical weights, Gaussian distributed inputs, Ising weights and inputs). For various cases, capacity ranges between more than 13 and 0.125, but as noted above it corresponds to recognition intrinsically associated with quantum measurement errors.

Finally, we would like to mention parameter κ, introduced by Gardner in her original paper Gardner [1988]. While strictly speaking, the classical perceptron checks whether the signal corresponding to a given input pattern has an appropriate (say positive) sign at the output, in principle one can demand that the signal is greater than a certain parameter κ. In the case of classical perceptrons with the scaling of connections proposed by Gardner, κ is N independent. In the more complex quantum case, we need to scale κ appropriately as in Eq. (2.2), and as discussed in the next section.

2.2.5 Storage capacity calculated using Gardner's program

As discussed in Chapter 1.5, the correct choice of the weights results in a desired input-output relation, i.e., a specific mapping between the input $\vec{i}^{\mu}$ and the output σ^{μ}. A learning rule is usually applied to find the correct weights, such as the Hebbian rule Sompolinsky [1987]. While the Hebbian rule has an appealing simplicity, Gardner, in her works Gardner [1988]; Gardner and Derrida [1988], was interested in the global properties of the classical perceptron model without specifying the learning rule. She asked the question: What is the maximum number of input-output patterns that the classical perceptron can realize? Therefore, she considered the relative volume in the space of possible weights, which realizes a given input-output relation.

Following Gardner's work, the storage capacity can be obtained from the fraction of $\vec{w}$-space which correctly reproduces the desired input-output relations normalized to the volume of vectors $\vec{w}$. When increasing the number of patterns, the volume of vectors $\vec{w}$ typically shrinks, and the relative volume of the weights vanishes. The limit of vanishing relative volume defines the storage capacity of the perceptron Nishimori [2001]. From the definition of the storage capacity the difference between the classical and quantum perceptrons results from the different dimensionality of the input vectors, which equals N physical inputs for the classical perceptron. In contrast, in the quantum case, the number of inputs equals the dimension of the Hilbert space m. Hence, for the classical perceptron we have $\alpha_c = p/N$, while for the quantum perceptron $\alpha_c = p/m$. For the classical perceptron, the storage capacity is known to be 2 and was calculated for example in Gardner [1988]; Gardner and Derrida [1988]; Müller *et al.* [1995]; Nishimori [2001].

The abundance of weights, which lead to desired input-output relations, can be treated by averaging over the weight vectors $\vec{w}$. This averaging gives rise to an ensemble of quantum machines, which can be analyzed with statistical physics tools. To define a finite volume of weights Gardner and Derrida [1988] we constrain the weight vector $\vec{w}$. Similar to Gardner's work one can consider two types of constraints: spherical weights, i.e., $|\vec{w}|^2 = m$ and Ising weights $w_i = \pm 1$. The corresponding integration measures Müller *et al.* [1995] are

$$\rho_S[\vec{w}] = \frac{1}{V_{S_0}} \delta \left(|\vec{w}|^2 - m \right) , \tag{2.4a}$$

$$\rho_I[\vec{w}] = \frac{1}{V_{I_0}} \prod_k [\delta (w_k - 1) + \delta (w_k + 1)] \tag{2.4b}$$

with the normalization (see Appendix B.3)

$$V_{S_0} = \int_w \delta(|\vec{w}|^2 - m) , \tag{2.5a}$$

$$V_{I_0} = \int_w \prod_k [\delta (w_k - 1) + \delta (w_k + 1)] . \tag{2.5b}$$

Then the relative volume of perceptrons, which fulfill a specific input-output relation, is given by

$$V_M = \int_w \prod_\mu \theta \left(\tfrac{1}{m} |\vec{i}^\mu \cdot \vec{w}|^2 - \kappa \right) \rho_M[\vec{w}] , \tag{2.6}$$

where the label $M = S$ for the spherical constraint or $M = I$ for the Ising constraint. The threshold κ takes values in $[0, m]$ and in the limit $\kappa \longrightarrow 0$ the relative volume allows us to obtain the maximum storage capacity of the quantum perceptron model Nishimori [2001]; Müller *et al.* [1995]. We calculate the relative volume using the integral representation of the Heaviside function

$$\theta (y - \kappa) = \int_\kappa^\infty d\lambda \int_{-\infty}^\infty \frac{dx}{2\pi} e^{ix(\lambda - y)} , \tag{2.7}$$

which we insert into Eq. (2.6). In the following we outline the calculation of the relative volume for the case of spherical weights and present the details of the calculation in App. B.

Spherical weights

The distribution of the spherical weights is given in Eq. (2.4a) and contains a delta function, which we represent via

$$\delta(|\vec{w}|^2 - m) = \int_{-\infty}^{\infty} \frac{dE}{2\pi} e^{iE\left(|\vec{w}|^2 - m\right)} . \tag{2.8}$$

Further, we average over the input vector $\vec{i}^\mu$ to avoid bias towards specific input vectors. The average with respect to $\vec{i}^\mu$ is denoted as $\langle\langle \cdot \rangle\rangle$. The expression for the relative volume becomes

$$\begin{aligned} \langle\langle V_S \rangle\rangle = \frac{1}{V_{S_0}} \int_w \int_\lambda \int_x \int_E \exp\left[iE\left(|\vec{w}|^2 - m\right)\right] \\ \times \langle\langle \exp\left[i \sum_\mu x^\mu \left(\lambda^\mu - \tfrac{1}{m} |\vec{i}^\mu \cdot \vec{w}|^2\right)\right] \rangle\rangle , \end{aligned} \tag{2.9}$$

where the integration measure is given in App. B.3.

Similar to Gardner we make the observation that Eq. (2.9) is a partition function of a classical spin glass, where $\langle\langle \cdot \rangle\rangle$ is interpreted as a disorder average and $\vec{w}$ is a classical spin variable. As for classical spin glasses Nishimori [2001]; Müller *et al.* [1995] we calculate $\langle\langle \log V_S \rangle\rangle$ via the replica trick

$$\langle\langle \log V_S \rangle\rangle = \lim_{n \to 0} \frac{\langle\langle V_S^n \rangle\rangle - 1}{n} , \tag{2.10}$$

which leads to the replicated variables $\vec{w}^\alpha$, x^α, λ^α with the replica index $\alpha \in \{1, \ldots, n\}$. Following the notation introduced in Nishimori [2001], we would like to emphasize that the reader should not confuse the storage capacity α with the replica index. In addition, we introduce the spin glass order parameter $q^{\alpha\beta}$ and its conjugate

$F^{\alpha\beta}$ via the integral

$$\begin{aligned} 1 &= \int_{-\infty}^{\infty} dq^{\alpha\beta} \delta\big(q^{\alpha\beta} - \tfrac{1}{m} \sum_k w_k^\alpha w_k^\beta\big) \\ &= m \int_{-\infty}^{\infty} dq^{\alpha\beta} \int_{-\infty}^{\infty} \frac{dF^{\alpha\beta}}{2\pi} e^{imF^{\alpha\beta}(q^{\alpha\beta} - \frac{1}{m}\sum_k w_k^\alpha w_k^\beta)} \end{aligned} \tag{2.11}$$

with $\alpha < \beta$. This identity is also referred to as Hubbard-Stratonovich transformation, see Sherrington and Kirkpatrick [1975]; Mezard *et al.* [1986] for details.

Ising inputs

In the next step, we perform the average over the inputs and assume small fluctuations of x^α, which leads to

$$\langle\langle V_S^n \rangle\rangle = \frac{1}{V_{S_0}^n} \int_F \int_q \int_E e^{mG}, \tag{2.12}$$

with integration measure given in App. B.3 and where we introduced the effective potential

$$G = \alpha G_1[q^{\alpha\beta}] + G_2[E^\alpha, F^{\alpha\beta}] - i \sum_\alpha E^\alpha + i \sum_{\alpha<\beta} F^{\alpha\beta} q^{\alpha\beta} \tag{2.13}$$

with the storage capacity α and the two contributions

$$\begin{aligned} G_1[q^{\alpha\beta}] = \log \int_{-\infty}^{\infty} \prod_\alpha \frac{dx^\alpha}{2\pi} \int_\kappa^\infty \prod_\alpha d\lambda^\alpha \\ \times \exp\left(i \sum_\alpha x^\alpha (\lambda^\alpha - 1) - \frac{1}{2} \sum_\alpha (x^\alpha)^2 - \sum_{\alpha<\beta} (q^{\alpha\beta})^2 x^\alpha x^\beta \right) \end{aligned} \tag{2.14}$$

and

$$G_2[E^\alpha, F^{\alpha\beta}] = \log \int_{-\infty}^{\infty} \prod_\alpha dw^\alpha \\ \times \exp\left(i \sum_\alpha E^\alpha (w^\alpha)^2 - i \sum_{\alpha<\beta} F^{\alpha\beta} w^\alpha w^\beta \right) . \tag{2.15}$$

Comparing the integrals for the effective potential reveals a quadratic dependency on $q^{\alpha\beta}$ for the quantum model and a linear dependence on $q^{\alpha\beta}$ for the classical model within the exponents. The non-linear dependence in the quantum case is a consequence of the measuring process, which involves the modulus square.

Note that $G = G(F, E, q; \alpha, \kappa)$ is a function of integration variables, and depends parametrically on α and κ. The integral over F, E, q in Eq. (2.13) can be evaluated using the saddle point method, due to the exponential dependence on m. In the replica symmetric case, one can eliminate dependence on E, F, so that the effective potential/free energy of interest can be defined in the limit n going to zero,

$$g(q; \alpha, \kappa) = \lim_{n\to 0} \frac{1}{n} G(q; \alpha, \kappa) , \tag{2.16}$$

It is useful also to introduce the proper normalization for the saddle point value of the effective potential, coming from subtracting the logarithm of $n \log(V_{S_0})$:

$$\tilde{g}(q_s; \alpha, \kappa) = g(q_s; \alpha, \kappa) - g(q_s; \alpha, \kappa = 0). \tag{2.17}$$

This quantity is strictly non-positive. If $\tilde{g}$ is one, then the volume is one (as it should happen for $\kappa = 0$). If $\tilde{g}$ is $-\infty$, then the volume shrinks to zero (as it should happen for $\alpha > \alpha_c$. This may happen even for $\kappa = 0$, due to the approximate character of our calculations. Finally, when $0 > \tilde{g} > -\infty$, the relative volume decreases exponentially with m as $\exp(m\tilde{g})$.

After the Hubbard-Stratonovich transformation we perform a saddle-point approximation for large m and assume replica symmetry, i.e.,

$$q^{\alpha\beta} = q\,, F^{\alpha\beta} = F\,, E^{\alpha} = E\,. \tag{2.18}$$

The saddle-point equations are

$$\frac{\partial G}{\partial E} = \frac{\partial G}{\partial F} = \frac{\partial G}{\partial q} = 0\,, \tag{2.19}$$

which we solve and subsequently perform the limit $n \rightarrow 0$.

Taking the derivative of G with respect to q and analyzing the limit $q \rightarrow 1$, we observe that it leads to the saddle point solution for q is $q = 0$ for $\alpha \leq \alpha_c(\kappa)$ to the maximum critical storage capacity at $\kappa = 0$ of

$$\alpha_{c,\text{max}} = 13.27 \geq 2. \tag{2.20}$$

In contrast, for $\alpha > \alpha_c(\kappa)$, q at the minimum of the effective potential becomes equal to 1, and the volume abruptly shrinks to zero. The saddle-point approximation allows us to study the critical storage capacity $\alpha_c(\kappa)$ as a function of the threshold κ, which we depict in Fig. 2.2. Note, the phase transition has here similar nature as in the classical perceptron. For $\alpha \leq \alpha_c(\kappa)$, the relative volume is equal to 1 for $\kappa = 0$. For $\kappa > 0$, the effective potential $\tilde{g} < 0$, and the volume shrinks moderately slowly exponentially with m as $\exp(mg)$. For $\alpha > \alpha_c(\kappa)$, $g = -\infty$, and the volume is strictly equal to zero (for details see Methods section).

Taking the derivative of G with respect to q and analyzing the limits $q \rightarrow 1$ and $\kappa \rightarrow 0$ leads to the maximum critical storage capacity of

$$\alpha_{c,\text{max}} = 4. \tag{2.21}$$

Additionally, the saddle-point approximation allows us to study the critical storage capacity α_c as a function of the threshold κ, which we depict in Fig. 2.2.

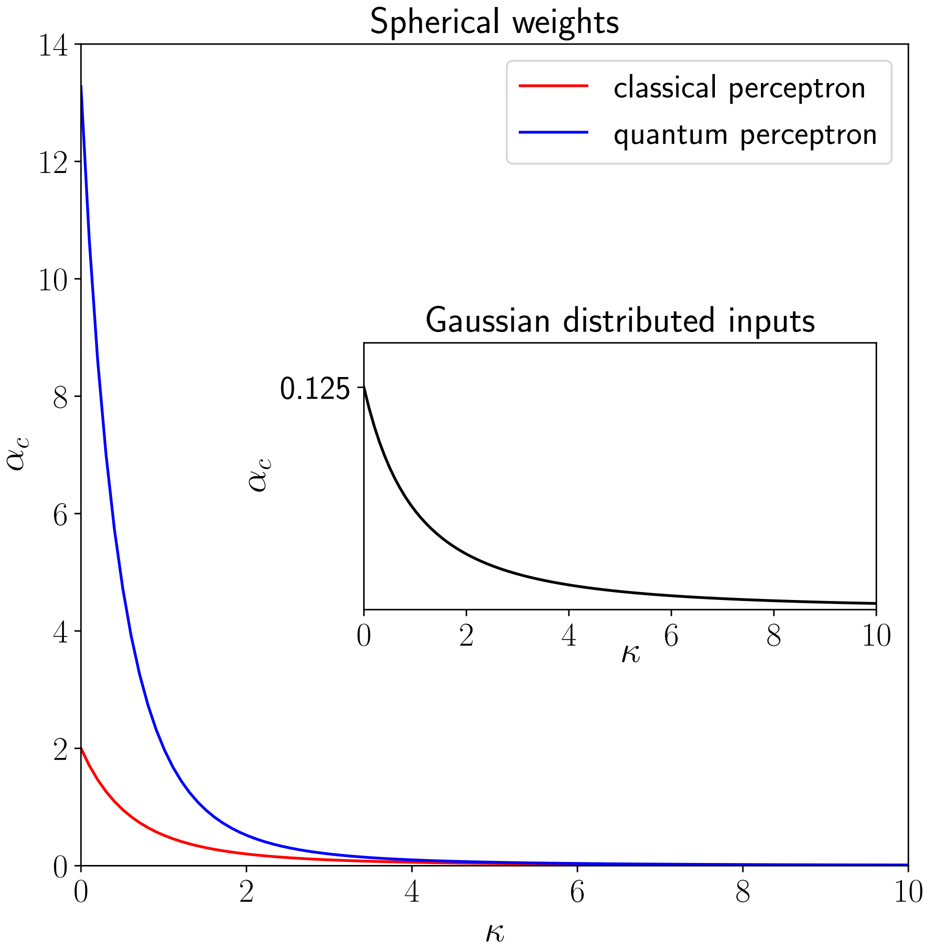


Figure 2.2 Storage properties of perceptrons for spherical weights: Storage capacity for the classical (red) and quantum (blue) perceptrons as a function of the threshold κ with Ising inputs. For $\kappa = 0$, the storage capacity has a maximum, whereas the storage capacity decays for $\kappa \gg 0$. In the inset, we plot the critical storage capacity as a function of κ for spherical weights, but with Gaussian distributed inputs.

Gaussian distributed inputs

In this section, we discuss the case of inputs distributed according to a Gaussian normal distribution. The weights in turn are distributed according to Eq. (2.4a). We apply the replica trick Eq. (2.10), introduce the order parameters as in Eq. (2.11) and average over the inputs to calculate the effective potential G. A comparison to Eq. (2.12) reveals that only the expression for G_1 changes

$$G_1[q^{\alpha\beta}] = \log \int_{-\infty}^{\infty} \prod_{\alpha} \frac{dx^\alpha}{2\pi} \int_{\kappa}^{\infty} \prod_{\alpha} d\lambda^\alpha \\ \times \exp \left[i \sum_{\alpha} x^\alpha \lambda^\alpha - \log \det \left(1 + 2i\hat{A} \right) \right], \tag{2.22}$$

where we introduce the matrix A later in Eq. (B.20). Next, we assume replica symmetry, i.e.,

$$q^{ab} = q, F^{ab} = iF, \tag{2.23}$$

where the imaginary unit i is used to assure that the saddle point solutions are real. Then the saddle-point equations are

$$\frac{\partial G}{\partial F} = \frac{\partial G}{\partial q} = 0, \tag{2.24}$$

which we solve and subsequently perform the limit $n \to 0$. The saddle-point equation given by the derivative with respect to q leads to

$$\alpha \left(2+\kappa\right)^2 q = \frac{q}{2\left(1-q\right)^2}. \tag{2.25}$$

This equation has one trivial solution $q = 0$ and one non-trivial in $0 < q < 1$. The non-trivial solution exists if and only if

$$2\alpha \left(2+\kappa\right)^2 \geq 1. \tag{2.26}$$

For $\alpha \leq \alpha_c = (1/2)(2+\kappa)^2$ the solution is trivial, and the logarithm of the relative volume is close to zero, it is proportional to $\tilde{g} = -\alpha\kappa/2$, and for $\kappa = 0$ the volume is equal to one, while for $\kappa > 0$ it decreases exponentially as $\exp(m\tilde{g})$ Above α_c, the saddle point solution for q is non-zero, and the volume shrinks also exponentially with m, but much faster (for details see Methods section). We plot α_c in Fig. 2.2 for different values of κ and observe $\alpha_c \rightarrow 1/8$ for $\kappa = 0$. The phase transition has a different character in comparison to Gardner's work Gardner [1988]. In her work, the volume decreases exponentially with m below the critical α_c (where $q < 1$), and strictly shrinks to zero above the critical α_c (where $q = 1$). In our work, the volume is close to one below α_c (although it decreases slowly exponentially with m, and it starts to decrease much more rapidly exponentially with m above α_c. This is the result of the approximations used (expansion in q). In App. B.5 we speculate, how one could to restore the "Gardner's nature" of the phase transition in our model with Gaussian inputs.

Ising weights

In the classical case the Ising weights were treated for example in Nishimori [2001]; Müller *et al.* [1995]; Gardner and Derrida [1988]. Here, we use the Ising weights for the quantum case and employ Eq. (2.4b) and Eq. (2.5) for the integration measure and normalization of the volume, respectively. We apply the replica trick Eq. (2.10), introduce the order parameters Eq. (2.11) and average over the inputs to calculate the effective potential. The contribution G_1 is the same as Eq. (2.15), while G_2 becomes

$$G_2[F^{\alpha\beta}] = \log \sum_{\{w^\alpha=\pm1\}} \exp\left(\sum_{\alpha<\beta} F^{\alpha\beta} w^\alpha w^\beta\right). \tag{2.27}$$

We assume replica symmetry, i.e.,

$$q^{ab} = q, F^{ab} = iF, \tag{2.28}$$

and the saddle-point equations are

$$\frac{\partial G}{\partial F} = \frac{\partial G}{\partial q} = 0. \tag{2.29}$$

Solving the saddle point equations in the limit for $q \longrightarrow 1$ we conclude that the storage capacity is

$$\alpha_c(\kappa) = \frac{4}{\pi}\left[\int_{-\kappa}^{\infty} Dy(\kappa + y)^2\right]^{-1}, \tag{2.30}$$

where we used the abbreviation

$$\int_{-\infty}^{\infty} Dy = \sqrt{\frac{1}{2\pi}}\int_{-\infty}^{\infty} dy\ e^{-\frac{y^2}{2}}. \tag{2.31}$$

In addition, we present the results of a Monte Carlo simulation in Fig. D.1, which shows that as $m \longrightarrow \infty$ and $\kappa \longrightarrow 0$, the storage capacity is 3.55 ± 0.01 (see App. B.4 for details). We interpret this discrepancy as the necessity for replica symmetry breaking. This analysis, however, goes beyond the scope of this paper.

2.2.6 Discussion

In this work, we calculated the storage capacity of a quantum perceptron proposed in a recent work Tacchino *et al.* [2019], which uses less memory resources compared to its classical counterpart (for N classical spins only $\log_2(N)$ are needed) and has already been implemented on IBM's quantum devices Tacchino *et al.* [2019, 2020a]. Following the seminal works of Gardner Gardner [1988]; Gardner and Derrida [1988], we use statistical physics techniques to calculate the storage capacity of this perceptron. In particular, we interpret this quantum perceptron as a classical perceptron on an extended input space with a different activation function, see Fig. 2.1. This interpretation allows us to calculate the storage capacity of a quantum

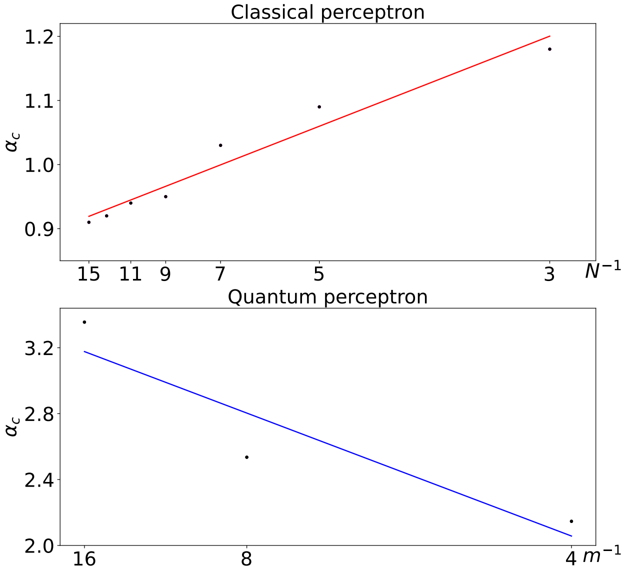


Figure 2.3 Storage properties of perceptrons for the Ising weights Storage capacity for the classical (red) and quantum (blue) Ising perceptrons as a function of the number of inputs N^{-1} and m^{-1}, respectively. The dots are the result of the Monte Carlo simulations. The intersection of the lines with the y-axis gives the storage capacity 0.86 ± 0.01 for the classical and 0.010 ± 0.005 for the quantum perceptron in the limits $N, m \rightarrow \infty$.

perceptron by computing the relative volume of quantum perceptrons which fulfill a given input-output relation.

To handle the multitude of inputs and learning rules, we integrate over the input and the weights. Formally, this averaging over input and weights maps the calculation of the relative volume to the partition function of a classical spin glass problem. Similar to problems in classical spin glass theory, we compute the logarithm of the partition function using the replica trick Nishimori [2001]; Müller *et al.* [1995]. Further, by using the large m expansion, we can determine the storage capacity α_c in leading order, which is the ratio of the stored patterns p over the computational resources m. Notably, the techniques presented here are applicable to other quantum architectures.

Given the model of Fig. 2.1c, we obtain a maximal critical storage capacity of $\alpha_{c,\max} = 4$ for the spherical weights, see Fig. 2.2. To put these results in perspective, we compare them with the classical perceptron. In the classical case, the maximal storage capacity is $\alpha_{c,\max} = 2$, see Fig. 2.2. Therefore, the maximum storage capacity of the quantum perceptron its double the maximum storage capacity of the classical perceptron. This result is in accordance with the recent work Abbas *et al.* [2021] where the authors found a similar relation to the capacity of quantum neural networks compared to the classical. Even though their definition of capacity is determined by the effective dimension, it also exploits the model's ability to express different relationships between variables. Moreover, taking into account that $\alpha_{c,\max}$ has a finite value for the quantum perceptron and that m equals the dimension of the Hilbert space ($m = 2^N$), the number of patterns that can be stored is exponential in the number of spins N in accordance with recent works Andrecut and Ali [2003]; Ventura and Martinez [2000]; Wright and McMahon [2020]. For example, in the recent work Lewenstein *et al.* [2021] the authors applied Gardner's program to attractor Quantum Neural Networks and found that the learning of P patterns at the order of $2^{N/2}$ is possible.

Given the model of Fig. 2.1c, we obtain a maximal critical storage capacity of $\alpha_{c,\max} > 13$ for the spherical weights and Ising inputs, see Fig. 2.2. To put these results in perspective, we compare them with the classical perceptron. In the classical case, the maximal storage capacity is $\alpha_{c,\max} = 2$, see Fig. 2.2. Therefore, the maximum storage capacity of the quantum perceptron is clearly larger than the maximum storage capacity of the classical perceptron. This result agrees with the recent work Abbas *et al.* [2021] where the authors found a similar relation to the capacity of quantum neural networks compared to the classical. Even though their definition of capacity is determined by the effective dimension, it also exploits the model's ability to express different relationships between variables.

For Gaussian distributed inputs, the performance of the quantum perceptron is quite different from the classical perceptron in accordance with a related work Benatti *et al.* [2022]. In the classical case, the relative volume shrinks exponentially with m below the critical capacity, and shrinks suddenly to zero above α_c, see Fontanari and Meir [1989]. In the present study, the volume shrinks exponentially with m, but the rate of shrinking changes from below ("easy learning" phase) to above α_c ("hard learning" phase). The maximum storage capacity of the "easy learning" phase is 0.125. These results suggest that the performance of quantum perceptron models does not always follow the behaviour of their classical counterparts. Therefore, it emphasizes the need to rigorously study these models and explore their properties as general learning machines.

Another example where contradictory behaviour is observed between classical and quantum perceptrons is in the case of Ising weights. The analytical results suggests that $\alpha_{c,\max} = 8/\pi$ while the Monte Carlo simulation suggests that $\alpha_{c,\max} = 3.55 \pm 0.01$. A similar discrepancy is observed in the classical perceptron with Ising weights as well. It is already known that the replica symmetry breaking is necessary to resolve this discrepancy for the classical perceptron Gardner and Derrida [1989a]; Nishimori [2001]. Therefore, a similar approach might be necessary to tackle the difference between the analytical and numerical result in the quantum case. We leave this open for future work.

Moreover it would be highly important for practical applications to explore the storage properties of the quantum perceptron with correlated inputs or input-output patterns. In a previous analysis of quantum perceptrons Lewenstein [1994], one distinguished between three different phases an *ignorant phase*, a *random phase*, and *learning phase*, and it would be interesting to detect these phases in the quantum perceptron architecture of Tacchino *et al.* [2019]. Future studies should also investigate the storage capacity away from $q \approx 1$ and the dependency on κ. Also, it will be essential to include corrections to the large m expansion and study the stability of the replica symmetric saddle point solution Mezard *et al.* [1986]. Finally, an exciting continuation of this work would be to consider other architectures of quantum perceptrons Schuld *et al.* [2015b]; Torrontegui and García-Ripoll [2019]; Gratsea and Huembeli [2021], e.g., qudit based platforms Kasper *et al.* [2020]; Weggemans *et al.* [2021]; Ringbauer *et al.* [2021] and analyze them with the tools presented in this work.

This work studied the storage properties of different quantum perceptron models with a direct hardware implementation Tacchino *et al.* [2019]. Importantly, inspired by the analysis of classical perceptrons Nishimori [2001], we applied statistical physics techniques of spin glasses to the studied quantum models. This also facilitated a certain comparison between the quantum and classical models, even though such a comparison is not perfectly sound. In particular, our work shows and validates that the number of patterns that can be stored in the considered models of quantum perceptrons is exponential in the number of spins N. We defined thus and calculated the corresponding values of the storage capacity for the studied quantum perceptrons as a ratio of the number of patterns p and the total number of spin configurations, $\alpha = p/m = p/2^N$.

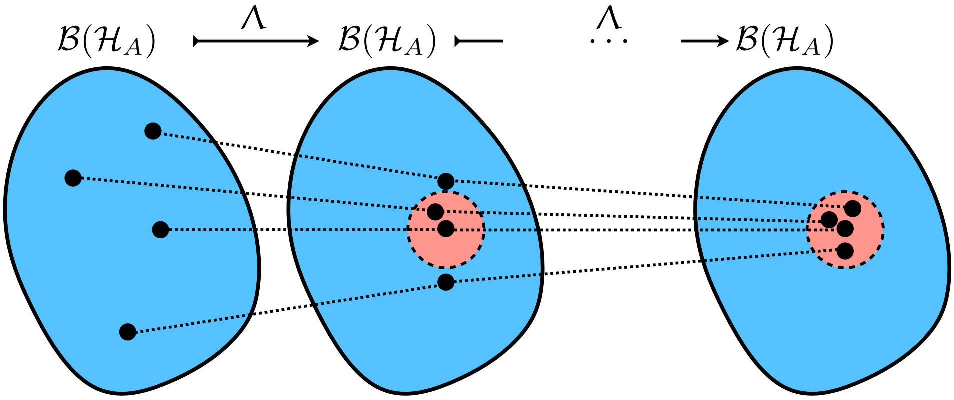

$\mathcal{B}(\mathcal{H}_A)$
Λ
$\mathcal{B}(\mathcal{H}_A)$
Λ
$\cdots$
$\mathcal{B}(\mathcal{H}_A)$

Figure 2.4 Color Online. Schematic representation of the action of CPTP maps Λ with N fixed states. Successive applications of $\Lambda : \mathcal{B}(\mathcal{H}_A) \mapsto \mathcal{B}(\mathcal{H}_A)$, brings arbitrary states $\rho \in \mathcal{B}(\mathcal{H}_A)$ to the set (depicted by red area) of stationary states of the map.

2.3 Gardner's program applied to attractor QNN

2.3.1 Storage capacity of attractor QNN

We associate QNNs with CPTP maps transforming initial states into final states in a finite (or infinite) time. Attractors (stored memory/patterns) correspond to the stationary states of the map, i.e., $\Lambda(\rho) = \rho$. We identify the storage capacity of QNNs (number of stored memories) with the maximal number of stationary points of CPTP maps acting on density matrices in N-dimensional Hilbert spaces. These maps act as attractors in the space of states, i.e., the successive application of the map brings an arbitrary state to the set of its fixed points, see Fig. 2.4. We interpret this class of maps as attractors QNNs (aQNNs).

To formalize the problem of the storage capacity of attractor QNN (aQNN), we consider an input (output) Hilbert space $\mathcal{H}_A$ ($\mathcal{H}_B$) of dimension N_A (N_B), and denote by $\mathcal{B}(\mathcal{H}_A)$ ($\mathcal{B}(\mathcal{H}_B)$) their respective sets of linear bounded operators. Quantum states ρ_A (ρ_B) correspond to positive semidefinite operators of unit trace in $\mathcal{B}(\mathcal{H}_A)$ ($\mathcal{B}(\mathcal{H}_B)$). Physical transformations can be characterized by CPTP maps, i.e., linear maps $\Lambda : \mathcal{B}(\mathcal{H}_A) \mapsto \mathcal{B}(\mathcal{H}_B)$ fulfilling: (*i*) positivity, $\Lambda(\rho) \geq 0$, $\forall \rho \geq 0$; (*ii*) complete positivity, that is, any extension of the form $(\mathcal{I} \otimes \Lambda) : \mathcal{B}(\mathcal{H}_C \otimes \mathcal{H}_A) \mapsto \mathcal{B}(\mathcal{H}_C \otimes \mathcal{H}_B)$ is also a positive map, where $\mathcal{I}$ is the identity map acting in an arbitrary space $\mathcal{H}_C$; and (*iii*) trace preservation, $\mathrm{Tr}[\Lambda(\rho)] = \mathrm{Tr}[\rho]$. A map Λ can be characterized by an Hermitian operator $E_\Lambda \in \mathcal{B}(\mathcal{H}_A \otimes \mathcal{H}_B)$, via the Jamiołkowski-Choi-Sudarshan (JCS) isomorphism **???**. For CPTP maps, the corresponding JCS operator is positive semidefinite $E_\Lambda \geq 0$ and fulfills (trace preserving condition) $\mathrm{Tr}_B[E_\Lambda] = \mathbb{1}_A$. The JCS operator reads $E_\Lambda = (\mathcal{I} \otimes \Lambda)(|\Omega\rangle\langle\Omega|)$, where $|\Omega\rangle = \sum_{i=1}^{N_A} |i\rangle|i\rangle$ is an unnormalized maximally entangled state in $\mathcal{H}_A \otimes \mathcal{H}_{A'}$, with $\mathcal{H}_{A'}$ a duplicate of the input space. In fact, this constitutes an isomorphism since $\Lambda(\rho_A) = \mathrm{Tr}_A[E_\Lambda \rho_A^T]$.

It is is well known that each CPTP map has, at least, one stationary state. Here we investigate which is the maximal number of linearly independent stationary states that non-trivial maps ($\Lambda \neq \mathcal{I}$) may have. In what follows, we address first this question assuming that stationary states correspond to projectors onto pure states. Unless specified, henceforth we take the Hilbert spaces dimensions to be $N_A = N_B = N$.

Theorem 1 There exist non-trivial CPTP maps Λ s.t. $\Lambda(|r_\mu\rangle\langle r_\mu|) = |r_\mu\rangle\langle r_\mu|$, where $\{|r_\mu\rangle\}$ are linearly independent and $\mu = 1, \ldots, N$.

Proof: 1) First, we transform Λ into a canonical form by noting that there always exist a linear, invertible transformation T, s.t. $|r_\mu\rangle = T|\mu\rangle$, where $\{|\mu\rangle\}_{\mu=1}^{N}$ form an orthonormal basis. Such transformation is unique up to : (a) the choice of the basis; (b) the phases of the basis elements that cancel in the projectors; and (c) the permutations of the elements of the basis. We define the canonical form $\tilde{\Lambda}$ as $\tilde{\Lambda}(\rho) = T^{-1}\Lambda(T\rho T^\dagger)(T^\dagger)^{-1}$, which has the property that if

$\tilde{\Lambda}(|\mu\rangle\langle\mu|) = |\mu\rangle\langle\mu|$ then $\Lambda(|r_\mu\rangle\langle r_\mu|) = |r_\mu\rangle\langle r_\mu|$.
2) We decompose the corresponding JCS operator as $E_{\tilde{\Lambda}} = E_{\mathcal{I}} + Q \geq 0$, where $E_{\mathcal{I}}$ is the JCS operator corresponding to the identity map. Then, non-triviality of $E_{\tilde{\Lambda}}$ requires that $Q \neq 0$. By definition $Q = Q^\dagger$, and $\mathrm{Tr}_B[Q] = 0$ (trace preserving). Moreover, for every $|\mu\rangle$ we have that $\langle\mu|Q|\mu\rangle = 0$, ergo $\langle\nu|\langle\mu|Q|\mu\rangle|\nu\rangle = 0$ for any μ, ν. This implies that $\langle\mu\nu|E_{\tilde{\Lambda}}|\mu\nu\rangle$=0 for $\mu \neq \nu$ and, because $E_{\tilde{\Lambda}} \geq 0$, then $E_{\tilde{\Lambda}}|\mu\nu\rangle = 0$ for $\nu \neq \mu$. As a consequence, $Q|\mu\nu\rangle = 0$ for $\mu \neq \nu$, which implies that Q has only nonzero matrix elements in the subspace spanned by the vectors $|\mu\mu\rangle$. Therefore, $Q = \sum_{\mu\nu} \alpha_{\mu\nu}|\mu\mu\rangle\langle\nu\nu|$, with $\alpha_{\mu\mu} = 0$. Finally, from $E_{\tilde{\Lambda}} \geq 0$, it follows that $|1 + \alpha_{\mu\nu}|^2 \leq 1$ for all $\mu \neq \nu$.

Such maps cause reduction of coherences in the orthonormal basis $\{|\mu\rangle\}$, namely if $\rho' = \Lambda(\rho)$, we find $|\langle\mu|\rho'|\nu\rangle| = |1+\alpha_{\mu\nu}||\langle\mu|\rho|\nu\rangle| \leq |\langle\mu|\rho|\nu\rangle|$ for $\mu \neq \nu$. The multiple iteration of these maps lead generically to a total decay of coherences (if for all $\mu \neq \nu$, $|1 + \alpha_{\mu\nu}| < 1$), and the memories stored will correspond to the fixed points of the dynamics (see Fig.2.4). Notice also that if $|1 + \alpha_{\mu\nu}| = 1$ for $\mu \neq \nu$, the state $|\varphi\rangle = a|\mu\rangle + b|\nu\rangle$ is also stationary. From the above theorem we immediately obtain:

Lemma 1 Any CPTP map Λ, s.t. $\Lambda(|\mu\rangle\langle\mu|) = |\mu\rangle\langle\mu|$, where $\{|\mu\rangle\}_{\mu=1}^N$ forms an orthonormal basis, has associated a JCS operator of the form:

$$E_\Lambda = \sum_{\mu}^{N} |\mu\mu\rangle\langle\mu\mu| + \sum_{\mu\neq\nu}^{N}(1 + \alpha_{\mu\nu})|\mu\mu\rangle\langle\nu\nu|, \tag{2.32}$$

with $\alpha_{\mu\nu} \in \mathbb{C}$ and $|1 + \alpha_{\mu\nu}| \leq 1$.

Here, we search for non-trivial maps i.e. neither the identity nor the unitary maps. Notice that unitary maps, and thus all evolution maps from a Hamiltonian, $\Lambda(\rho) = U\rho U^\dagger$ are the extreme case in which $1 + \alpha_{\mu\nu}$ are the phase factors $e^{i(\varphi_\mu - \varphi_\nu)}$, where $e^{i\varphi_\mu}$ are eigenvalues of U. The convergence speed to the attractors is determined by $|1 + \alpha_{\mu\nu}|$ and the initial state.

Theorem 2.3.1. *Since a non-trivial Λ exists for $M = N$, there are even more such maps for $M < N$.*

Theorem 2.3.2. *If $M \geq N + 1$, that is, $\Lambda(|\mu\rangle\langle\mu|) = |\mu\rangle\langle\mu|$ for $\mu = 1, \cdots, N$ and $\Lambda(|e\rangle\langle e|) = |e\rangle\langle e|$, where $|e\rangle = \sum_{\mu=1}^{N} c_\mu |\mu\rangle$ with all $c_\mu \neq 0$, then the map is trivial, $\Lambda \equiv \mathcal{I}$.*

This implies that the non-trivial CPTP maps with $M \geq N + 1$ stationary pure states ceases to exist if for all $\mu \neq \nu$, $|1 + \alpha_{\mu,\nu}| < 1$, which is, generically, the case.

2.3.2 Calculation of the relative volume using Garnder's program

The total volume of aQNN corresponds to the volume of CPTP maps that have a fixed set of stationary states. The volumes of various sets of maps have been estimated in **??** using the JCS isomorphism. The approach used by Szarek *et al.* **??** estimates, using the Hilbert-Schmidt norm, the radius of a ball that approximates the volume of CPTP maps in the asymptotic limit. In this limit, $N \to \infty$, this radius is $R = \exp(-1/4)$. The manifold of CPTP maps acting on a Hilbert space of dimension N, has dimension $d = N^4 - N^2$, which corresponds to the dimension of the space of Hermitian JCS matrices (N^4), minus the number of real constraints imposed by trace-preserving condition (N^2). Using the volume of the unit ball, in the limit of sufficiently large N, the volume of the CPTP manifold approximates **?** to:

$$V_{\text{CPTP}}(d) = \frac{\pi^{d/2}}{\Gamma(d/2 + 1)} \exp(-d/4). \tag{2.33}$$

An estimate for the volume of the aQNN manifold with exactly M stationary linearly independent pure states, is obtained by imposing M conditions in the d-dimensional space of the CPTP manifold:

$$V_{\text{aQNN}}(\epsilon, M, d) = \int d^d V_{\text{HS}} \prod_{\mu=1}^{M} \mathbf{1}_{[1-\epsilon]} \left(\langle \mu\mu | E_\Lambda | \mu\mu \rangle\right), \tag{2.34}$$

where $d^d V_{\text{HS}}$ is the Hilbert-Schmidt measure **?** and $\mathbf{1}_{[1-\epsilon]}(x)$ is the indicator function, being one for $x \in [1-\epsilon]$ and zero otherwise. The parameter ϵ defines a basin of attraction. By definition, V_{aQNN} must be smaller than V_{CPTP}. For sufficiently small ϵ, we may approximate $\mathbf{1}_{[1-\epsilon]}(x) \simeq \epsilon\delta(x-1)$. In this case, the integral over the d-dimensional manifold of CPTP maps with the M constraints reduces from d to $d-M$ dimensions. For $1 \ll M \ll N$ but still large d, the radius of the corresponding ball remains asymptotically the same, and the volume of CPTP maps with M stationary states (see Fig. 2.5) becomes

$$V_{\text{aQNN}}(\epsilon, M, d) \simeq \frac{\epsilon^M \pi^{(d-M)/2}}{\Gamma((d-M)/2+1)} e^{-(d-M)/4}. \tag{2.35}$$

Note that this result does not depend on the concrete choice of the stationary states $\{|\mu\mu\rangle\}$. Notice also that, as shown in Lemma 1, there are infinitely many CPTP maps with M stationary states, but their volume is of measure zero for $\epsilon = 0$.
The relative volume reads then:

$$V_R(\epsilon, M, d) = \frac{V_{\text{aQNN}}(\epsilon, M, d)}{V_{\text{CPTP}}(d)} \simeq \frac{\epsilon^M e^{M/4} \pi^{-M/2} (d/2)!}{((d-M)/2)!}. \tag{2.36}$$

Using Stirling's formula we obtain:

$$\ln V_R(\epsilon, M, d) \simeq \frac{M}{2} \ln\left(\frac{\sqrt{e} d \epsilon^2}{2\pi}\right) - \frac{M^2}{4d}. \tag{2.37}$$

The choice of the parameter ϵ should be sufficiently small in order to be consistent with $V_R(\epsilon, M, d) < 1$, which follows from Eq. (2.34).

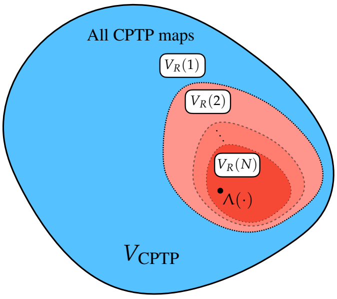


Figure 2.5 Color online. Representation of the relative volume $V_R(M)$ of CPTP maps acting as aQNN and storing M stationary pure states. The volume shrinks as we increase the number of stationary states from $V_{\text{CPTP}} = V_R(1)$ for $M = 1$, to $V_R(N)$ for $M = N$.

Since we are interested in the scaling with M, an upper bound corresponds to setting $\epsilon = e^{-1/4}\sqrt{(2\pi)/d}$. Then the relative volume scales as

$$V_R(M, d) \simeq \exp(-M^2/4d), \tag{2.38}$$

shrinking surprisingly slowly with M. Hence, the learning of $M \ll N$ patterns should be feasible for aQNNs. In particular, for systems of n qubits where $N = 2^n$, M can be of order $2^{n/2}$, that is, exponential in the number of qubits.

In what follows, we generalize our results to the case where the fixed points (stored memories) correspond to mixed states. To this aim we introduce the so-called *classical ensembles* as defined recently by Kronberg **?**.

Definition 1 Let $\mathcal{E} = \{\rho_\mu\}$ with $\mu = 1, \cdots, M$ be an ensemble of N-dimensional density matrices in $\mathcal{B}(\mathbb{C}^N)$. The ensemble $\mathcal{E}$ is called *classical* if there exists a single invertible operation T that diagonalizes all elements of the ensemble; i.e., $T\rho_\mu T^\dagger = D_\mu$, where all D_μ are simultaneously diagonal. We call this basis the *computational basis*.
The above definition generalizes the one given in **?**, since in our case T does not have to be unitary. Although the maximal number of

linearly independent density matrices in $\mathcal{E}$ equals N, the ensemble may contain many more elements, $M \geq N$.
Theorem 1′ There exist non-trivial CPTP maps Λ, s.t. $\Lambda(\rho_\mu) = \rho_\mu$, where $\rho_\mu \in \mathcal{E}$ with $\mu = 1, \ldots, M$, and arbitrary M.

Proof: The ensemble is determined by the complete basis in which all elements are diagonal. The required map, up to the canonical transformation to the corresponding orthonormal basis $\{|\mu\rangle\}$, has the form given by Eq. (2.32). Since $\Lambda(|\mu\rangle\langle\mu|) = |\mu\rangle\langle\mu|$ for all $|\mu\rangle\langle\mu|$, then $\Lambda(\rho_\mu) = \rho_\mu$ is also true for any ρ_μ that is a mixture of projectors $|\mu\rangle\langle\mu|$ and, therefore, for all the members of $\mathcal{E}$.

The relative volume of the aQNN for the ensemble $\mathcal{E}$ behaves, however, differently that in the case of storing linearly indepedent pure states. Now, having $M \leq N$ stationary mixed states $\Lambda(\rho_\mu) = \rho_\mu$ for $\mu = 1, \cdots, M$, demands imposing $\sim M \times N^2$ constrains in the d-dimensional space of the CPTP manifold. In turn, this means that the relative volume of CPTP maps storing M mixed states should behave approximately as $V_R(M, d) \sim e^{-M^2}$, decreasing very rapidly with M.

Storage capacity of feed-forward QNN The generalization of the above results to feed-forward QNNs is presented in the Appendix. There, we consider the case corresponding to different input and output dimensions.

2.3.3 Discussion

As already emphasized, one crucial feature of NNs is their storage capacity for associative memory, that is, the number of *patterns* (stored memories/attractors) the network has for a given number of neurons n. For attractor NNs (aNNs) of the Hopfield-type Hopfield [1982], the relevant question is to determine how many stationary states, serving as stored memories, the network may have. In standard Hopfield models, where neurons are Ising spins and attractors correspond to metastable states resulting from two-body spin-spin interactions, the storage capacity scales $\sim O(n)$. Recently, it has been shown that if the Hopfield model is extended to p-body interactions, in the so-called Dense Associative Network model Hopfield and Tank [1986], the capacity of storage can be highly increased, surpassing the linear $O(n)$ behaviour and reaching $O(n^{p-1}/\ln n)$, or even beyond **?**.

Preliminary attempts to analyze the storage capacity of QNNs were pursued in Lewenstein [1994]. In a different approach, an exponential increase of the storage capacity for a specific quantum search algorithm was demonstrated in Ventura and Martinez [1998]. More recently Rebentrost *et al.* [2018]; Meinhardt *et al.* [2020], an increased storage capacity was obtained by using a feed-forward interpretation of quantum Hopfield NNs. Similarly, qudits have been studied in the context of quantum machine learning **?**Beer *et al.* [2020]. One crucial question in this context is whether the storage capacity of QNNs offers a quantum advantage.

In this work we addressed this question by associating QNNs to CPTP maps transforming initial states into final states in a finite (or infinite) time. We demonstrated that there exist a family of (non-trivial) CPTP maps that have $M = N$ linearly independent stationary pure states, and provide the generic expression of such maps. We demonstrated, using CPTP maps acting on a Hilbert space of dimension N, that aQNN's can store up to N linear independent pure states. For n qubits, quantum channels reach thus the capacity 2^n, clearly outperforming the storage capacity of standard classical neural networks $\sim O(n)$, where n is the number of binary neurons, or the best Dense Associative Networks whose storage capacity $\sim O(2^{(n/2)})$.

We also analyzed the learning capability of QNN's by applying Gardner's program to the quantum case and estimate the relative volume of QNNs realizing the desired attractor input-output relations. We related the learning capability of aQNN's to the relative volume $V_R(M)$ of CPTP maps with M stationary pure states, and show, in the limit of large N, that this volume decreases very slowly with the number of stored patterns M as $\exp(-M^2/(N^4 - N^2))$. Our results signal quantum advantage meaning that CPTP maps acting on n-qubit states may reach a storage capacity of $\mathcal{O}(2^n)$, surpassing the storage capacity of classical neural networks, including Dense Associative Networks.

Next, we derived analogous results for bilayer QNNs and their respective attractor input-output relations. Finally, we applied our procedure to feed-forward QNN with different input and output spaces. Our results are simple and mathematically rigorous. Furthermore, they open the path to study the relation between the storage capacity of QNNs and the quantum features, such as coherence and entanglement, of the desired attractor input-output relations.

2.4 Conclusions

Both our works on quantum perceptrons and quantum neural networks suggest that we are still in search of good models for quantum machine learning. We believe that we could take inspiration of how classical perceptrons and neural networks were explored- by distilling information of how the neurons in our brain work instead of trying to mimic their behaviour. Similarly, we need to distill the information of how classical perceptrons and neural networks work when we investigate how to implement their quantum version. Furthermore, we introduce tools to describe these models and explore their properties. As we move forward, more fundamental research is necessary to further bridge this gab. This is essentially a good way to spend our time until fault-tolerant quantum computation will be available to perform a large scale benchmarking of QML models Schuld and Killoran [2022b].

Chapter 3

Exploring QP and QNN structures with a teacher-student scheme

In recent years different implementations of quantum perceptron (QP) and quantum neural networks (QNN) have been proposed on quantum hardware. Even though the plethora of the quantum circuit structures that emerge, tools that could directly benchmark them are missing. In this work, we address this question by introducing a systematic tool that could be used to directly benchmark different quantum models.

Specifically we introduce the teacher-student scheme which has also been applied to the training and benchmarking of classical models. The main idea is that each quantum model plays once the role of the teacher and once the role of the student. Therefore this method helps to explore the quantum model's performance in terms of its training capability. This in turn allows to look for a potential quantum advantage of a certain quantum circuit structure over others.

In this chapter we introduce the teacher-student scheme that could be used for benchmarking quantum models even when large scale quantum computing simulations could be realized.

3.1 Introduction

In the previous chapter we explored the quantum advantage that QP and QNN have over their classical counterparts in terms of their storage properties by applying Gardner's program. Here we introduce numerical tools to further explore the performance of such quantum models in terms of the models' ability to learn through training. The QP and QNN are defined through variational quantum circuits (VQC) Peruzzo *et al.* [2014a]; Grimsley *et al.* [2019]; Babbush *et al.* [2014]; Lewenstein [1994]; Schuld *et al.* [2014] which have attracted a lot of interest for applications on NISQ quantum devices and is one of the leading models for implementing quantum machine learning. An increasing number of literature tries to understand the potential power of quantum models as variational circuits.

To this end we introduce the teacher-student scheme as a numerical tool to compare the relative expressive power of different quantum models and explore a potential quantum advantage of a specific QP or QNN structure over the increasing number of such quantum models that are introduced in the literature. With the teacher-student scheme Torrey and Taylor [2013]; Ba and Caruana [2014]; Hinton *et al.* [2015], the studied quantum models are introduced once as a teacher and once as a student. This way we can avoid to generate synthetic datasets that might give an advantage to certain architecture structures. Therefore, the teacher-student scheme creates a fair framework for comparing any quantum models. Moreover we use different realizations of the teachers and obtain the average performances for the students. And thus, this scheme offers a systematic comparison which, for example, could help further understand the role of the hidden units and non-linearities of quantum models.

The chapter is organized as follows: In Section 3.2 we introduce the QP and QNN as variational quantum circuits inspired by recent works Tacchino *et al.* [2019]; Pérez-Salinas *et al.* [2020]. In Section 3.3 we introduce the teacher-student scheme which allows for a systematic comparison between different quantum models and can explore which quantum models have a potential quantum advantage in terms of their expressive power. In Section 3.4 we compare the QP and QNN along with some alterations of these models to better understand the role of the hidden units and the non-linearities. Finally, Section 3.5 contains the conclusions and outlook.

The original results presented in this chapter are based on the publication [Gratsea and Huembeli, 2021] with P. Huembeli.

3.2 Quantum Perceptrons as variational quantum circuit models

Different works define quantum perceptron and QNN models as variational quantum circuits (VQC): Tacchino *et al.* [2020b]; Pérez-Salinas *et al.* [2020]; Sharma *et al.* [2020]; Cong *et al.* [2019]. Following this approach, for a given data set $\mathcal{D} = \{(\boldsymbol{x}^k, y^k)\}$, a quantum circuit $U(\boldsymbol{x}^k, \boldsymbol{w})$ is parameterized for every input data point $\boldsymbol{x}^k$ with label y^k and with trainable parameters $\boldsymbol{w}$. The initial state is $|0\rangle^{\otimes N}$, where N is the number of qubits. We denote $|\psi^k\rangle = U(\boldsymbol{x}^k, \boldsymbol{w})\,|0\rangle^{\otimes N}$ as the output state of the quantum perceptron. To do classification with only two labels we measure one qubit in a single direction, e.g. in Z direction and interpret the expectation value $\langle\psi^k|\,Z\,|\psi^k\rangle$ as the label prediction. To train the quantum model, we define the cost function

$$C = \sum_k (y^k - \langle\psi^k|\,Z\,|\psi^k\rangle)^2, \tag{3.1}$$

which is minimized during the training. Here, we choose a two dimensional input data set with $\boldsymbol{x}^k = (x_1^k, x_2^k)$ and encode the data to the circuit with the gates $U(\boldsymbol{x}) = R_x(x_1) \otimes R_x(x_2)$ applied on two distinct qubits, where $R_x(\phi) = \exp(-i\phi\sigma_x/2)$ is the single qubit X rotation gate. These data encoding gates are depicted in light and dark orange colors respectively in the circuit diagrams (e.g. Figure 3.1) and we will refer to it as the angle encoding. Throughout this work, we use the same encoding for all models to have a fair comparison. The trainable part of the perceptron is realized by parameterized single qubit rotations $Rot(\phi, \theta, \omega) = R_z(\omega)R_y(\theta)R_z(\phi)$ and controlled-Z gates. At the output of each QP we apply a multi-controlled NOT gate inspired by Tacchino *et al.* [2019]. The trainable gates are depicted in blue colors in the figure and we will refer to them as processing gates.

Here we focus on two models, the QP and the RU quantum models. The main advantage of the QP is its simplicity and analogy to the classical perceptron Tacchino *et al.* [2019], while the RU has attracted a lot of interest recently and has shown great success Schuld *et al.* [2020b]; Pérez-Salinas *et al.* [2020].

3.2.1 Quantum Perceptron (QP)

The QP is inspired by the recent work of Tacchino et. al. Tacchino *et al.* [2019]. In their work, the quantum perceptron closely resembles the binary valued classical perceptron. The input is a m-dimensional real vector $\boldsymbol{x}$ with binary values ±1 for each element x_i. The input quantum state $|\psi^k\rangle = U(\boldsymbol{x}^k)|0\rangle^{\otimes N}$ is realized by an encoding gate $U(\boldsymbol{x}^k)$ acting on N qubits. The output state of the perceptron is defined as $U(\boldsymbol{w})|\psi^k\rangle$, where $U(\boldsymbol{w})$ is a parameterized unitary. A multi-controlled NOT gate activates the ancillary qubit which is measured to obtain the quantum perceptron output. The weights $\boldsymbol{w}$ are chosen such that the ancillary qubit is activated with a probability $p(|1\rangle_a) = |\sum_i^m x_i w_i|^2$. One can set a threshold for the expectation value $\langle Z \rangle$ of the ancilla qubit to obtain a binary output from this probability.

To build deep structures from QPs one can, like for their classical analog, use the output of one perceptron as the input of the next Tacchino *et al.* [2020a]. Therefore, for a hidden QP, the input state $|\psi^k\rangle$ comes from the ancillary qubit of the previous perceptron and the hidden perceptron itself consists of a parameterized processing gate $U(\boldsymbol{w})$, a (multi)-controlled NOT gate that activates another ancillary system and a deferred measurement. This architecture allows one to build deep structures in a coherent way and the ancillary systems can be measured at the very end of the circuit. The idea is that because of this deferred measurement the required non-linearity is still introduced between the layers.

In their original work Tacchino *et al.* [2019], the authors focus on binary valued inputs and weights and later they extended it to a continuous values Mangini *et al.* [2020]. Both methods though need the basis encoding which requires many qubits to encode the data. To avoid this, we use the angle encoding introduced earlier in this section. This will now play the role of the gate $U(\boldsymbol{x}^k)$, while the parameterized gates $Rot(\phi, \theta, \omega)$ (in blue in the circuit diagrams) together with the CZ entangling gates realize the $U(\boldsymbol{w})$ gates for the data processing. This allows for continuous inputs and weights. The perceptron model for the simple case of a two dimensional input is illustrated in Fig. 3.1a.

3.2.2 Re-Uploading Architecture (RU)

The RU is inspired by the recent work of Perez-Salinas et. al. Pérez-Salinas *et al.* [2020]. In their work, they propose a quantum version of a ffNN with only one hidden layer. To emulate the behaviour of a ffNN, they repeatedly apply data encoding unitaries $U(\boldsymbol{x}^k)$ parameterized with the input data $\boldsymbol{x}^k$ to the circuit. Each of these data dependent unitaries is followed by a parameterized unitary $U(\boldsymbol{w})$ with trainable parameters $\boldsymbol{w}$. Because of the repeated application of $L_i = U(\boldsymbol{w})U(\boldsymbol{x})$ to the circuit, this scheme is called data re-uploading. A single application of L_i is often referred as a re-uploading layer and their number determine the architecture of the re-uploading circuits.

In this work, we use the angle encoding as defined earlier and we parameterize the processing gates with $R(\phi, \theta, \omega)$ and CZ gates. To simplify the measurement procedure and make the two perceptron models more comparable, we also use a multi-controlled NOT gate to activate an ancilla qubit exactly like in QP. We also use the same

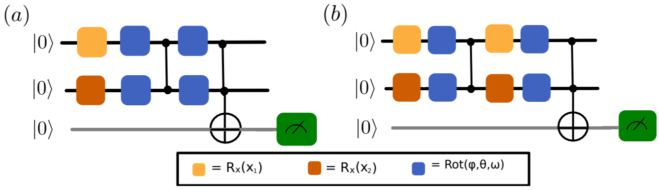


Figure 3.1 The quantum circuits that represent the quantum perceptron (QP) (a) and re-uploading quantum model (RU) (b).

cost function from Eq. (3.1).

The notion of depth is not as clearly defined for the re-uploading scheme as it is for the QP. It is known that more layers of re-uploading lead to better expressivity of the model Schuld *et al.* [2020b], but there is no clear classical analog to this. Therefore, we will compare the RU with an increasing number of re-uploading layers L_i to QNN models with an increasing number of QPs. In the next section, we elaborate the scheme that we use to compare the aforementioned architectures.

3.3 Teacher-student scheme

Here, we introduce the teacher-student scheme that aims to to benchmark different realizations of QP and QNN against each other. We systematically compare the two aforementioned architectures (QP and RU see Fig.3.1), but the scheme could be used for any circuits. The main idea is that one architecture (for example the QP) will play the role of the teacher and generate the labels that will be used to train the student (for example the re-uploading quantum model). Thus, we avoid to generate artificial data sets, such as e.g the circle data set in Pérez-Salinas *et al.* [2020], that could possibly favour one of the architectures. With the teacher-student scheme, we directly see the data structures that each architecture can generate and how well other architectures could learn them.

3.3.1 Notion of teacher

The teacher generates the labels for a fixed set of inputs $\mathcal{D} = \{\boldsymbol{x}^k\}$ with 2 dimensional input vectors $\boldsymbol{x} = \{x_1, x_2\}$ on a grid $x_i \in [-\pi, \pi]$. For a fixed teacher architecture we choose several random initializations for the parameters $\boldsymbol{w}$ of the processing gates (blue squares in the figures). We use the measurement outcomes of the ancilla qubit as the model predictions y^k of the input data. This way we can generate several data sets for different random initializations. The predictions have continuous values $y^k \in [-1, 1]$ given by the outcome of the measurement $\langle \psi^k | \, Z \, | \psi^k \rangle$, but we also generate binary valued labels by choosing $y^k_{\text{binary}} = \text{sign}(y^k)$. The teachers with binary labels focus more on the basic characteristics of the data structures, while the ones with continuous labels also care for the details. We can visualize the data structures with *prediction maps*, which are the density plots of the model predictions and labels y^k for the input data $\boldsymbol{x}^k$.

3.3.2 Notion of student

We train the students with the labeled data generated by their teachers to learn those data structures. It is not obvious how to define a good/bad student, since different tools can be used to characterize their performance. The *prediction maps* of the students are best for visualizing the similarity of the student's and teacher's predictions of the label y^k to gain qualitative results. For a more rigorous quantitative comparison we compute the *relative entropy* between the student's and teacher's outputs y^k. Specifically, we use the information divergence (Kullback–Leibler divergence or relative entropy) which defines a distinguishable measure between two probability distributions P and Q **?**:

$$S(P \| Q) = \sum_{i=1}^{N} p_i \ln \frac{p_i}{q_i}. \tag{3.2}$$

When the two distributions are similar, the value of the relative entropy is close to zero. To interpret the predicted labels y^k as probabilities, we offset and re-normalize them ($y^k > 0$, $\sum_{\boldsymbol{x}^k \in \mathcal{D}} y^k = 1$). Then, to compare two prediction maps, the information divergence is calculated by summing over the whole input space. Here, we are interested in the average relative entropy of all teacher-student pairs. Another qualitative metric is the *loss function* which determines the success of the training. When the student is trained with the binary valued labels the percentage of the correctly predicted labels can be computed. We refer to this as the *accuracy score* which gives an overall performance of the student. To identify a good/bad student all these tools should be taken into account.

3.4 Results

3.4.1 Toy model

Here, we will elaborate the notion of the teacher-student scheme with an illustrative example. We use the RU architecture as a teacher (Fig. 3.1b) and both, QP and RU, as students (Fig. 3.1a and b). A very characteristic example for the prediction maps is shown in Fig. 3.2, where the Student - QP does not learn the inner structure (circles) at the left and right sides of the prediction map of the teacher. The Student - RU reproduces almost perfectly the teacher as expected, since they have the same circuit architecture. But the Student RU finds a different set of parameters compared to the one of the Teacher RU. We present the parameters of the processing gates for both of them in the Appendix (C.1.1). The solid lines in the “Loss curve" in Fig.3.2 show the loss of the two students that correspond to these particular prediction maps, where Student - Re-uploading achieves a much lower loss. If we use the binary labels of the teacher and train again the students we reach the accuracy score approximately equal to 0.9 compared to 0.8 for the QP. Both students have a high accuracy score, since the topology of the prediction maps of the students and

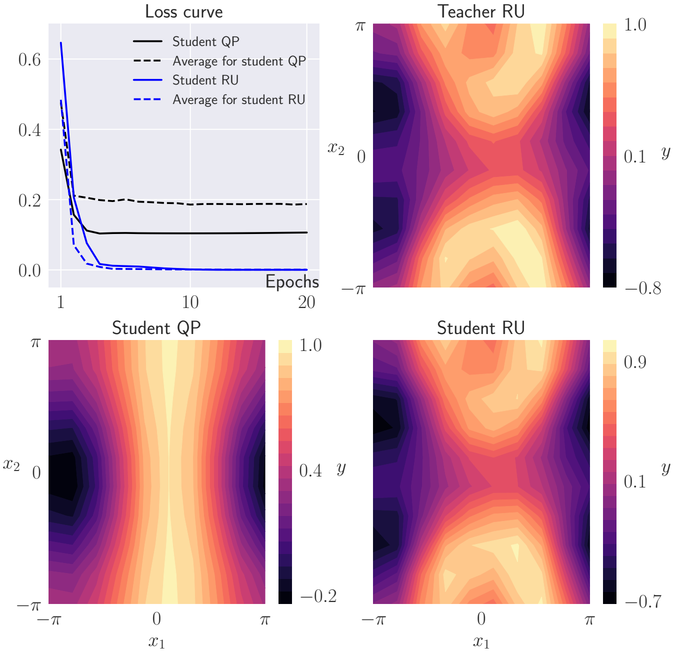


Figure 3.2 Teacher-student training for the teacher RU (Fig. 3.1b) and 2 students (QP and RU in Fig. 3.1 a and b, respectively). The prediction maps show one particular example of the training with the corresponding losses (solid loss curves). The average loss (dashed lines) show the average over all 10 random initializations. The teacher-student training for the teacher QP can be found in the appendix Fig.C.3.

the teacher with binary labels are very similar (see Appendix C.1.2, Fig. C.2).

We generated 10 different data sets by randomly initializing the teacher's parameters as explained earlier to obtain the average performance of the student. Both students are trained on those data sets and we characterize their performance with the tools mentioned earlier. The dashed lines in Fig. 3.2 show the average of the loss function over all the initializations of the teacher for both students. The Student - RU shows good convergence (blue dashed line) for all data sets generated by the teacher, since the teacher and student have the same architecture in this Toy example. The average loss for the Student - QP (black dashed line) is larger than the single example shown in the prediction map (black solid line), which suggests some of the teacher's data sets can be learned more accurately by the student than others. These results are also supported by the calculation of the average relative entropy over all prediction maps which is equal to 0.247 and 0.001 for the Student - QP and the Student - RU, respectively. It is worth noting here that the loss captures better the global differences in the prediction (i.e. a general off-set of the whole prediction maps), while the relative entropy captures local differences in the maps (i.e. the local minima that do not appear in one of the maps). For more details, we provide the explicit code used for this toy model in **?**.

In order to have a complete comparison of the two models, we now use the QP architecture as the teacher and train again for both students (see Fig. C.1 in Appendix). The average loss for both students converges to similar low values (see Fig. C.1 in Appendix) and we have a high accuracy score 0.9 of both students. The average relative entropy is approximately equal to zero for both students (0.0002 for the QP and 0.0013 for the RU). These results suggest that the QP generates simpler data structures and both architectures can learn them.

In conclusion, the RU architecture can generate and learn more complex output distributions than the QP. It can learn all the data sets generated by the QP with almost zero loss and relative entropy. On the contrary, the QP is not able to learn the data sets provided by the RU. Even for this toy example, all four metrics defined earlier were taken into account, which reveals the difficulty on how to determine whether a student is good or bad. Therefore, one should take under consideration all the different tools introduced earlier (the prediction maps, the accuracy score, the (average) loss and relative entropy) to characterize the performance of the students.

3.4.2 A more complex teacher

Next, we compare the behaviour of the two models in Fig. 3.1 (Students) to a “deep" re-uploading architecture in Fig. 3.3 (Teacher) with many repetitions of the encoding gates. In Fig. 3.3, we show a characteristic result for the prediction maps. The Student - RU learns the basic features of the teacher’s architecture contrary to the Student - QP which learns a very simplified version of the data structure. As mentioned earlier, the number of encoding unitaries determine the function complexity that can be learned Schuld *et al.* [2020b]. Therefore, the teacher with four pairs of encoding gates generate a more complex data structure compared to the ones that the student of the QP and RU can learn with one and two pairs of encoding unitaries, respectively. If we train the students on the binary valued outputs, we reach accuracy scores approximately equal to 0.8 for both students, because the topology of the prediction maps of the students and the teacher with binary labels are very similar (see Appendix C.1.2 Fig. C.3). Therefore, the accuracy score does not allow to determine which student learns the teacher more accurately. But the prediction maps for this characteristic example (see Fig. 3.3) shows the better performance of the Student - RU. Even more, the loss curves, and specifically the average loss curves, support this claim. As we can see from Fig. 3.3, the average loss curve for Student - RU (blue dashed line) over 10 realizations of the teacher is much lower than the average loss curve of Student - QP (black dashed line).

Finally, the average relative entropy is equal to 0.15 and 0.11 for the Student - QP and the Student - RU, respectively and enhances this statement.

3.4.3 Improving the QP

Here, we present some alterations of the Student - QP to improve its performance and further understand its behaviour. So far, we used four general single qubit rotations and two 2-qubit gates (the multi-controlled NOT included). Sousa et al. **?** show that eight single qubit unitaries and four entangling gates realize an arbitrary two qubit gate. Therefore, we add four more processing gates to the Student - QP each followed by CNOT entangling gates (see figure B.1a). We still have a single perceptron with two inputs, but with a more general unitary for processing the data. The performance of the perceptron though did not improve, suggesting that the structure of the circuit is not enough to simulate the data structure of the re-uploading perceptron as a teacher.

Next, we try to increase the complexity of the QP by realizing a deep QNN from single QPs inspired by Tacchino *et al.* [2020a]. The structure is shown in Fig. B.1b with the classical analog in the inset which can be understood in the following way: The input data (light and dark orange circles in the inset) are introduced with the encoding unitaries (light and dark orange squares). The processing happens with the blue unitaries which correspond to the blue circles in the inset. The output of each blue circle is realized with the CNOT gate acting on both ancilla qubits 1 and 2. These outputs are then introduced in the last ancilla qubit 3 with the CZ gates. The performance of the circuit did not improve though (see Fig. B.2), which means that the deferred measurement on the second ancilla is not enough to improve the performance of the architecture. This shows that the formation of coherent deep QNN structures does not follow the classical analog of a classical deep NN, i.e. by staking several perceptrons together.

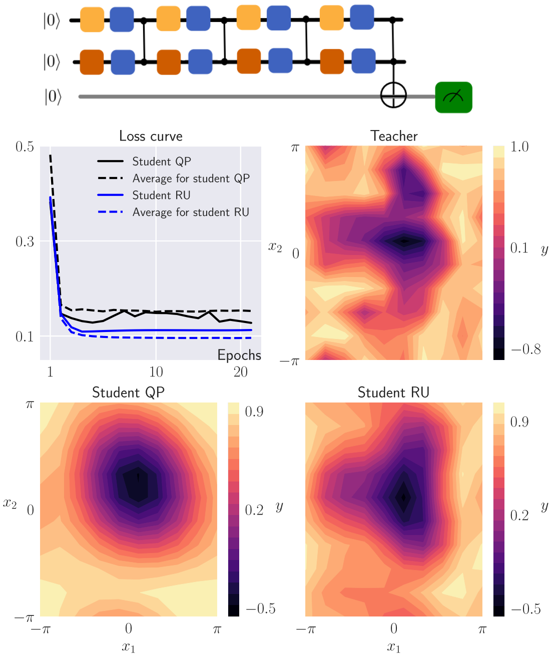


Figure 3.3 The “deep" architecture of the teacher in terms of the number of times the data are encoded. The prediction maps of the teacher with the deep architecture and the students: Student - QP and Student - RU as in Fig. 3.1. In the upper right corner, we plot the loss curves for the Student - QP (black line), Student - RU (blue line) and the student averages over 10 different realizations of the teacher (Student - QP: black dashed line and Student - RU: blue dashed line).

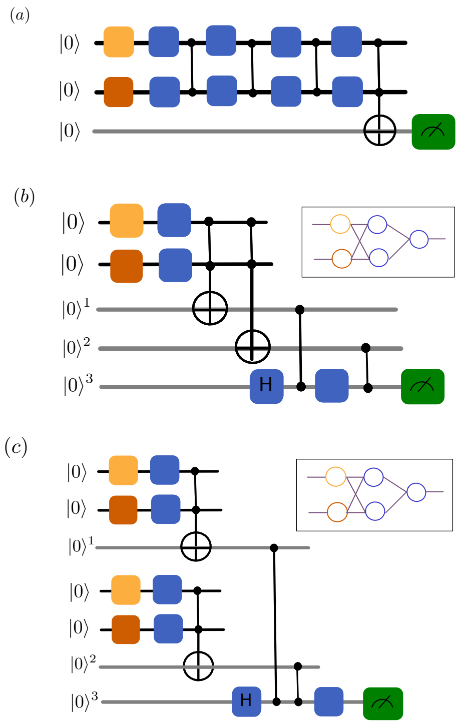


Figure 3.4 The alterations of the Student - QP to an eight gate QP (a), a QNN model (b) and a quantum ffNN as proposed in Tacchino *et al.* [2020a]. In the insets, the analogy to classical deep NN is shown for the circuits in (b) and (c). The models in (a) and (b) have the same expressivity with the QP, contrary to the quantum ffNN (c) which performs better as the RU (see Fig. B.2).

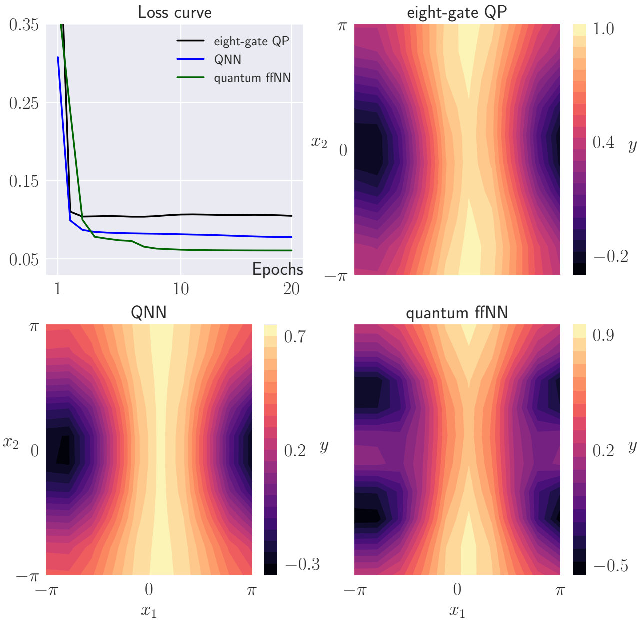


Figure 3.5 The prediction maps of the students in Figures B.1a, B.1b and B.1c and their loss curves.

Another alteration of the QP is shown in Fig. B.1c. There we use the QNN structure that Tacchino et. al. proposed in Tacchino *et al.* [2020a], where the QP is used as a building block to create the QNN, exactly like in the classical case. Specifically, instead of using the output of one QP twice as in Fig. B.1b, we use two QP to form a QNN. This circuit shows improved learning capability and performs quite well as suggested by the convergence to a low value of the loss curve (see Fig.B.2). Therefore, comparing the QNN (Fig. B.1c) with the deep QP (Fig.B.1b) suggests that the deferred measurement does not provide the necessary depth to the circuit and shows the importance of re-uploading the data several times to increase the performance of the architecture.

Even though the QNN architecture in Fig. B.1c seems quite complex (many qubits and entangling gates), at the end is very similar to the circuit architecture of the RU (Fig. 3.1b). Both circuits encode the data twice in a vertical or in a horizontal way, respectively. For the vertical case $2N$ qubits are used for N encodings of the data (see Fig. B.1c), while in the horizontal case the N encodings apply in the same two qubits (see Fig. 3.1b). The recent work of Schuld et. al. Schuld *et al.* [2020b] shows that the complexity of the functions that a circuit learns is determined by the number of times that the encoding gates are applied. They emphasize that there is no difference if the encoding data are introduced in a circuit vertically (as in the QNN Fig. B.1c) or horizontally (as in the RU Fig. 3.1b). In that sense, the architectures of these two models are equivalent and the prediction maps enhance this claim. But, as suggested in the work Schuld *et al.* [2020b], the trainable circuit and measurement could affect the functions that can be approximated. We already see here the qualitative difference of these two models. For the QNN in Fig. B.1c, the average accuracy score is equal to 0.84 and the average relative entropy gives 0.098 which is significantly different from the values of the RU in Fig. 3.1b (0.9 and 0.0009, respectively). This suggests that the architecture Fig. B.1c has similar performance to the RU, but as the metrics suggest there are some qualitative differences between them due to the differences at the trainable part of the circuits Schuld *et al.* [2020b].

3.4.4 Role of the encoding

Contrary to classical perceptrons, quantum perceptrons can have different encodings, i.e. amplitude, basis and angle encoding **?**. In this work, the encoding of the two-dimensional data $\boldsymbol{x} = (x_1, x_2)$ is achieved with the application of $R_x(x_1)$ and $R_x(x_2)$ to two distinct qubits. A good choice of the encoding can immediately lead to a linear separation of the data as suggested in recent works **??**. Therefore, exploring a quantum perceptron model is interconnected with exploring the encoding of the input data. Here, we want to compare the aforementioned angle encoding with the encoding $Rot(x_1, x_2, 0)H$ applied at each qubit. The Hadamard gate H is applied to $|0\rangle$, since otherwise for $Rot(\phi, \theta, \omega) = R_z(\omega)R_y(\theta)R_z(\phi)$, the first gate $R_z(x_1)$ applied to $|0\rangle$ would not contribute anything.

To see how important the encoding is to separate a predefined data set, we apply the encoding for all data $\boldsymbol{x}^k$ from a circular data set (see Fig. C.5 in Appendix) without the processing unitaries and generate the states $|\psi^k\rangle = U(\boldsymbol{x}^k)\,|0\rangle = c_{00}\,|00\rangle + c_{01}\,|01\rangle + c_{10}\,|10\rangle + c_{11}\,|11\rangle$ for both encodings. Specifically, we have $|\psi^k\rangle_{RX}$ for $U(\boldsymbol{x}^k) = R_x(x_1) \otimes R_x(x_2)$ and $|\psi^k\rangle_{Rot}$ for $U(\boldsymbol{x}^k) = Rot(x_1, x_2, 0) \otimes H \otimes Rot(x_1, x_2, 0)H$. We find the probability vectors $P = [p_{00}, p_{01}, p_{10}, p_{11}]$ for each of the states $|\psi^k\rangle$, with $p_{i,j} = |c_{i,j}|^2$. We are interested in the probability vectors and not the quantum states themselves because for our perceptron models, the probability p_{11} corresponds to the activation probability of the Toffoli gate. Specifically, we encode 500 input data points $\{x_1, x_2\}$ from the circular data set and create the probability vectors for each one of them for the different encodings. We apply principle component analysis (PCA) on each one of the probability vectors which converts them to two dimensional data sets. Then, we can see whether the data are already separable just after the encoding. In Fig.3.6, we plot these two dimensional data sets for both encodings. With the RX encoding the data are already separable, contrary to the Rot encoding where the data are along the same curve. The data encoded with RX, shown in Fig.3.6(left), can be separated with a quadratic function as discussed in the work **?** and a single general parameterized unitary $U(\boldsymbol{w})$ that post processes the state $|\psi^k\rangle$ would suffice. The data in 3.6(right) cannot be separated by a single parameterized processing unitary and would need for example more data re-uploadings.

The aforementioned results stress the importance of the encoding in a quantum circuit Schuld *et al.* [2020b] and shows that synthetic data sets can favour certain architectures. If the data are already separated after the encoding gates, the parameterized gates will simply rotate the data along the measurement axis. But, if the encoding fails to separate the data as in the case of Rot, the parameterized gates will not be able to separate them on their own. Also, we want to emphasize that the re-normalization of the data affects the success of the training, since the quantum functions are periodic and the input data should lie within the period of the function **?** (see Appendix C.1.4 Fig. C.6). Finally, it worths mentioning that the encoding of the data labels could also affect the success of the training (see Appendix C.1.3 Fig. C.5).

3.5 Discussion

Inspired by the recent works Pérez-Salinas *et al.* [2020]; Tacchino *et al.* [2019], we explored the expressive power of QPs, their formation to QNNs and the RU models implemented on NISQ devices. In order to systematically compare the architectures, we introduced a so-called teacher-student scheme, where the studied models are introduced once as a teacher and once as a student. This way we can avoid to generate synthetic data sets that might give an advantage to certain architectures and it creates a more fair framework for comparing any quantum models.

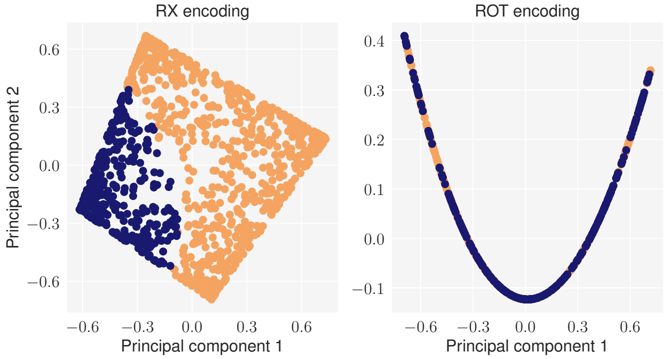


Figure 3.6 PCA on the probability distribution for the RX (left) and Rot (right) encodings. One can see that the circuit data immediately separates for the RX encoding but it does not for the Rot.

Specifically, we showed that the deep structures that can be built with QPs only increase the expressivity of a model if the data are uploaded several times. It is not sufficient to use deferred measurements to generate hidden non-linearities similar to classical NNs if the output of a QP is reused. We explored several different ways of how to leverage deferred measurements to generate hidden non-linearities, but the expressive power of QNNs only improved when additional data-uploadings were added (Fig. B.1c). This suggests that the non-linear behaviour induced by a measurement of a single QP cannot be generalized to deep QNNs if the single QPs are cast together in a coherent way (Fig. B.1b). Therefore, it is still an open question how to build deep QNNs in a coherent way, where measurements only occur at the end of the computation. Thus, one should not expect a one-to-one mapping of quantum and classical NNs.

These results are in accordance with the recent work **?**, which shows that the number of times that the data are encoded determines the functions that can be approximated. The needed non-linearities in a quantum model can be generated (apart from the measurement) from the encoding gates that are non-linear functions of the input data. Performing PCA on the probability vectors, we showed that given the encoding, the data can already be separated without further processing. Therefore, the performance of a QP is strongly affected by the encoding and the dataset itself. Apart from the encoding, the processing plays an important role as well. The universal approximation capability of different quantum models has been discussed extensively in Pérez-Salinas *et al.* [2020]; Schuld *et al.* [2020b]; Pérez-Salinas *et al.* [2021a,b], but it does not provide any information about how well the circuit could perform or how many parameters it needs to approximate a function within a certain error. The calculation of the average relative entropy for Fig.B.1c and Fig.3.1b showed that the trainable part of the circuit affects the functions that can be approximated. This effect will be further explored in subsequent research.

For future work, it will be of great interest to explore different perceptron models and compare their performance with the teacher-student scheme. Then, the question arises which perceptron model will be the ideal building block of QNN architectures and how quantum perceptrons could be combined to form a deep QNN. Another research direction is to explore other quantum models with no direct analog with classical NN, like the re-uploading model or quantum kernels in general. Finally, it would be of great importance to further explore the role of entanglement and encoding in QPs, in their formation to QNNs and in other quantum models.

3.6 Conclusion

This work is a first example of systematic tools that could be used to benchmark different QP and QNN architectures. Moving forward the teacher-student scheme could be applied to benchmark the training capabilities of quantum models on different problem instances. Therefore, this work set the foundation from where to explore a potential quantum advantage in terms of the training capabilities of different quantum models.

Chapter 4

The performance of quantum models

In recent years a plethora of Quantum Machine Learning (QML) models are introduced as parameterized quantum circuits Tacchino *et al.* [2019]; Torrontegui and García-Ripoll [2019]; Peruzzo *et al.* [2014a]. In this work, we extend the tools that could be used to explore their performance. The performance could be defined as the model's ability to learn through training Pérez-Salinas *et al.* [2020]; Schuld *et al.* [2020b], but also the model's ability to extract information from the input state to the readout qubit Schuld *et al.* [2020b]; Sim *et al.* [2019a].

To this end, we introduce relevant tools that could quantify the performance and help us better understand how different elements affect the overall performance of the studied model Schuld *et al.* [2020b]; Casas and Cervera-Lierta [2023]. We focus on small circuit structures in order to better understand the effect of the different elements to the overall performance, but also to clarify how to use these tools in practise.

This work is a stepping stone to better understand how to explore the performance of quantum models that could potentially result in a quantum advantage.

4.1 Introduction

In the previous chapter we introduced the Teacher-student scheme as a tool to compare the performance of different QP and QNNs architectures as VQC and explored a potential quantum advantage in terms of the model's ability to learn through training. In this chapter, we shift the focus from QP and QNN models to arbitrary quantum machine learning (QML) models, which consist of an encoding, processing and measurement part. Specifically, the QML models are defined as VQC, since the processing part is parameterized. We focus on small quantum circuits with a few number of qubits due to many open questions: ”How many parameterized gates should be used?”, ”How many entangling gates and where?”, ”Where should the circuit be measured?” and ”Do ancilla qubits help improve the expressivity of the circuit?”.

Inspired by these questions, in this chapter we focus on extending the numerical tools used to analyze a potential quantum advantage of QML models in terms of their performance. Here the term performance characterizes the model's ability to learn through training Benedetti *et al.* [2019a], the type of functions that it can express Schuld *et al.* [2020b] and its scrambling capability Wu *et al.* [2021]; Shen *et al.* [2020], i.e. extracting information from the input state to the readout qubit. Specifically, we extend the TS scheme introduced in chapter 3 to further characterize the model's ability to learn through training. We also apply the averaged operator size that characterizes quantum information scrambling. Finally, we use the representation of quantum models as Fourier series to explore the function classes that quantum circuits "have access to".

The original results presented in this chapter are based on the publication Gratsea and Huembeli [2022] with P. Huembeli.

4.2 Tools for analysing QML models

We study quantum circuits that consist of three parts: the encoding unitaries $S(x)$, the processing gates U_θ (consisting of data independent but parameterized gates) and the measurement operator M. The building blocks of such arbitrary quantum circuits are shown in Fig. 4.1. Throughout this work, we focus on supervised learning tasks with a given data set $\mathcal{D} = \{(\boldsymbol{x}^k, y^k)\}$, where the input data $\boldsymbol{x}^k$ are encoded by $S\left(\boldsymbol{x}^k\right)$ and y^k are the labels given by the measurement outcome. The initial state is $|0\rangle^{\otimes N}$, where N is the number of qubits. As shown in Fig.4.1, we first apply the encoding unitary $S(\boldsymbol{x})$ (yellow), then the processing U_θ (blue) and finally the measurement operator M (green).

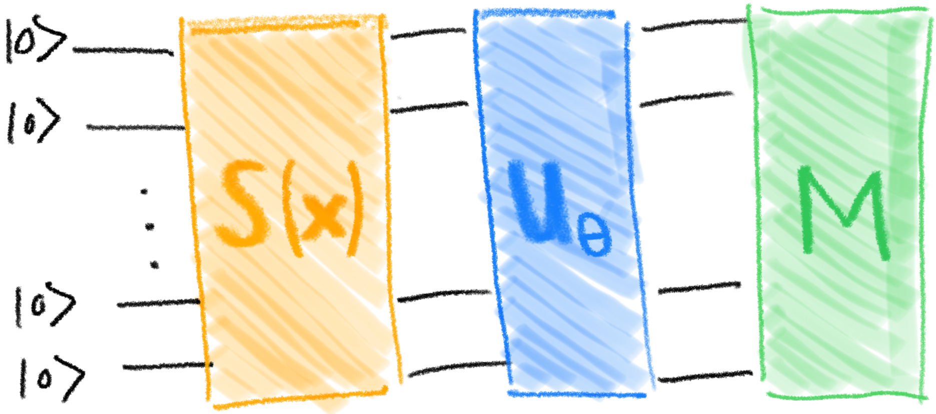


Figure 4.1 The building blocks for an arbitrary quantum circuit: encoding (yellow unitary), processing (blue parameterized unitary) and measurement (green operator).

In this section, we focus on two theoretical tools for characterizing the performance of QML models, i.e. the averaged operator size Wu *et al.* [2021] and the Fourier representation Schuld *et al.* [2020b]. These methods give us quantitative results on the effect of the processing and measurement operators (blue parameterized unitary and green operator in Fig.4.1). We also employ an extended version of the teacher-student scheme discussed in the previous chapter to quantify differences in the prediction maps numerically, i.e. the density plots of the model predictions or labels y^k for the input data $\boldsymbol{x}^k$.

4.2.1 Averaged operator size

In the works Wu *et al.* [2021]; Nahum *et al.* [2018]; Roberts *et al.* [2018], the authors introduce related quantities to characterize quantum information scrambling, i.e. extracting information from the input state $S(\boldsymbol{x})|\mathbf{0}\rangle$ to the readout qubit. In the work of Wu *et al.* [2021], the authors propose the averaged operator size - a quantity that depends on the circuit architecture, the processing and the measurement operators. Moreover, its value is positively correlated with the learning efficiency of the QNN architecture.

To calculate the averaged operator size of an arbitrary operator $\hat{O}$, we decompose it into a summation of a Pauli strings, which can be done for any Hermitian operator Nielsen and Chuang [2011]. Such a decomposition has the form

$$\hat{O} = \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}} \hat{\sigma}^1_{\alpha_1} \otimes \hat{\sigma}^2_{\alpha_2} \cdots \otimes \hat{\sigma}^n_{\alpha_n}, \tag{4.1}$$

where $\hat{\sigma}^i_{\alpha_i}$ for $\alpha_i \in \{0, 1, 2, 3\}$ define the Pauli operators including the identity acting on qubit i. The coefficients c_α can be computed by

$$c_\alpha = \frac{1}{2^n} \mathrm{Tr}\left(O \hat{\sigma}^1_{\alpha_1} \otimes \hat{\sigma}^2_{\alpha_2} \cdots \otimes \hat{\sigma}^n_{\alpha_n} \right). \tag{4.2}$$

The operator size of a Hermitian operator $\hat{O}$ is given by

$$\mathrm{Size}(\hat{O}) = \sum_{\boldsymbol{\alpha}} |c_{\boldsymbol{\alpha}}|^2 \, l(\boldsymbol{\alpha}), \tag{4.3}$$

where $c_{\boldsymbol{\alpha}}$ are the coefficients of decomposition in Eq. (4.1) and $l(\boldsymbol{\alpha})$ counts the number of non-identity matrices in each Kronecker product in the summation given by the same equation. In our case, the hermitian operator $\hat{O} = \hat{U_\theta}^\dagger \hat{M} \hat{\mathcal{U}}_\theta$ depends only on the processing unitary U_θ and the measurement M operators.

To obtain the averaged operator size for a given circuit architecture U_θ and a fixed measurement M, we take the average over Haar random unitaries U_θ

$$\overline{\mathrm{Size}} = \int d\hat{U}_\theta \,\mathrm{Size}\left(\hat{U_\theta}^\dagger \hat{M} \hat{U}_\theta\right). \tag{4.4}$$

A larger value of the average operator size suggests a more expressive circuit structure Wu *et al.* [2021].

4.2.2 Map differencies from teacher-student scheme

The teacher student scheme introduced in Chapter 3 can be used to compare performance of quantum models with three different quantitative scores were considered: the accuracy score, the (average) loss and the relative entropy and the prediction maps for a more qualitative overview. Here, we add another score to the TS scheme that directly compares differences between prediction maps which allows us to better quantify them. Specifically, we compute the average of the differences between the prediction maps of the teacher and student

$$\overline{\Delta y} = \frac{1}{p}\sum_{k}^{p} |y_T^k - y_S^k|, \tag{4.5}$$

where $\{y_T^k\}$ are the generated labels from the teacher and $\{y_S^k\}$ the learned labels of the student where k runs over all points p of a given input dataset $\{x^k\}$. Also, y_T^k, y_S^k take continuous values in $[-1, 1]$, but for the calculation of the $\overline{\Delta y}$ we re-scale them to be in $[0, 1]$. We can present the $\overline{\Delta y}$ as a percentage difference between the two studied models. This allows us directly to compare how similar the prediction maps of the two models are. It is a quantitative measure of how well the student can learn the teacher, i.e. a $\overline{\Delta y}$ equal to zero suggests that the student learns the teacher perfectly, while high values for $\overline{\Delta y}$ indicate that the student is not able to learn the labeling provided by the teacher.

4.2.3 The representation of quantum models with partial Fourier series

In the work of Schuld *et al.* [2020b], the authors explore how the data encoding influences the class of functions that a quantum model can learn. Circuit architectures employed in supervised tasks with multiple encoding unitaries on different qubit can be mapped to a partial multivariate Fourier series

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{\boldsymbol{j}} \sum_{\boldsymbol{k}} c_{\boldsymbol{jk}} e^{i\boldsymbol{x}\cdot(\boldsymbol{\lambda_k} - \boldsymbol{\lambda}_j)}, \tag{4.6}$$

where $j, k \in \left[2^d\right]^N$ with N the number of qubits and d is the dimension of an encoding gate, i.e. $d = 1$ if it is a single qubit gate. As it is explained in the original work, the frequency spectrum determines the functions that the quantum model could express, while the coefficients $c_{\boldsymbol{jk}}$ determine how the accessible functions can be combined.

The number of Fourier basis functions is solely determined by the eigenvalues of the data-encoding Hamiltonians. This means that repeated data encoding gives rise to a larger frequency spectrum Ω and more complicated function classes. The processing and the measurement of the circuit determine the coefficients, and therefore, how the accessible functions can be combined. Inspired by the work of Schuld *et al.* [2020b], in Section 4.3.3, we study how the processing and measurement operators of simple quantum circuits affect the coefficients, and therefore, the final function classes that these quantum circuits "has access to". This analysis sheds light on how individual elements of the processing and measurement architecture affect the function classes that the quantum model can express.

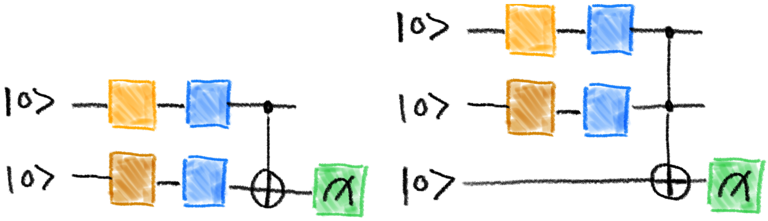


Figure 4.2 A 2-qubit and 3-qubit simple circuit architecture. The light and dark orange colors represent the data encoding gates, while the blue the parameterized gates. A CNOT gate is applied before the measurement operator for the 2-qubit circuit, while a Toffoli gate for the 3-qubit circuit.

4.3 Simple quantum models

In this section, we apply the tools described in the Section 4.2 on two simple quantum models shown in Fig. 4.2. We use a two dimensional input data set with $\boldsymbol{x}^k = (x_1^k, x_2^k)$ and encode it to the circuit with the gates $S(\boldsymbol{x}) = R_x(x_1) \otimes R_x(x_2)$ applied on two distinct qubits, where $R_x(\phi) = \exp(-i\phi\sigma_x/2)$ is the single qubit X rotation gate. These data encoding gates are depicted in light and dark orange colors respectively in the circuit diagrams. The blue gates represent the parameterized single qubit rotations $Rot(\phi, \theta, \omega) = R_z(\omega)R_y(\theta)R_z(\phi)$. The predictions take continuous values $y^k \in [-1, 1]$ given by the outcome of the measurement $\langle\psi^k| Z |\psi^k\rangle$, where $|\psi^k\rangle = U_\theta S(\boldsymbol{x}) |0\rangle^{\otimes N}$.

4.3.1 Averaged operator size of simple quantum models

We compute the averaged operator size of Eq. (4.4) for the 2-qubit and 3-qubit circuits of Fig. 4.2 by taking the average over the Haar random unitaries for the parameterized single qubit unitaries depicted with blue color. For the numerical simulation we use a Monte-Carlo integration McClean introduced in the Appendix D.1. In Fig.4.3, we plot the averaged operator size as a function of the number of parameterized gates. The error bars are the standard deviation from the mean value. We increase the number of parameterized gates by adding a layer L with two arbitrary single qubit gates (blues color) and an a CNOT entangling gate, $L = CNOTRot(\phi, \theta, \omega) \otimes Rot(\phi', \theta', \omega')$. We stress here that the 3-qubit circuit always has a Toffoli gate before the measurement operator.

The average operator size for the 2-qubit case has approximately the same value around 1.60 independent of the number of gates used (see blue solid line of Fig. 4.3). The drop observed from two to four parameterized gates is small while the error is large. Therefore, this suggests that for the 2-qubit case the performance is approximately the same as we increase the number of parameterized gates. On the contrary, for the 3-qubit case, the averaged operator size significantly increases once we move from two to four parameterized gates, and afterwards, reaches a plateau. These results already suggest that the 3-qubit case performs better than the 2-qubit case, i.e. the circuit structure of the processing and measurement operator is more expressive.

Next, we focus on the maximum value of the operator size, which corresponds to specific values of the parameterized angles that give the best possible performance of this circuit structure (represented with the dashed line in Fig. 4.3). From the dashed lines in Fig. 4.3, we see that for both circuits the maximum value of the operator size is equal to 2 for the circuits with two parameterized gates. As we increase the number of parameterized (and entangling gates), the maximum value of the operator size is increased for the 3-qubit, but quickly reaches a plateau. Once again, these results suggest that the 3-qubit circuit structure is more expressive than the 2-qubit.

4.3.2 Teacher-student scheme

To further analyze the expressive power of the two models in Fig. 4.2, we employ the Teacher-student scheme introduced in Section 4.2.2. The analysis of the differences $\overline{\Delta y}$ of the prediction maps validates

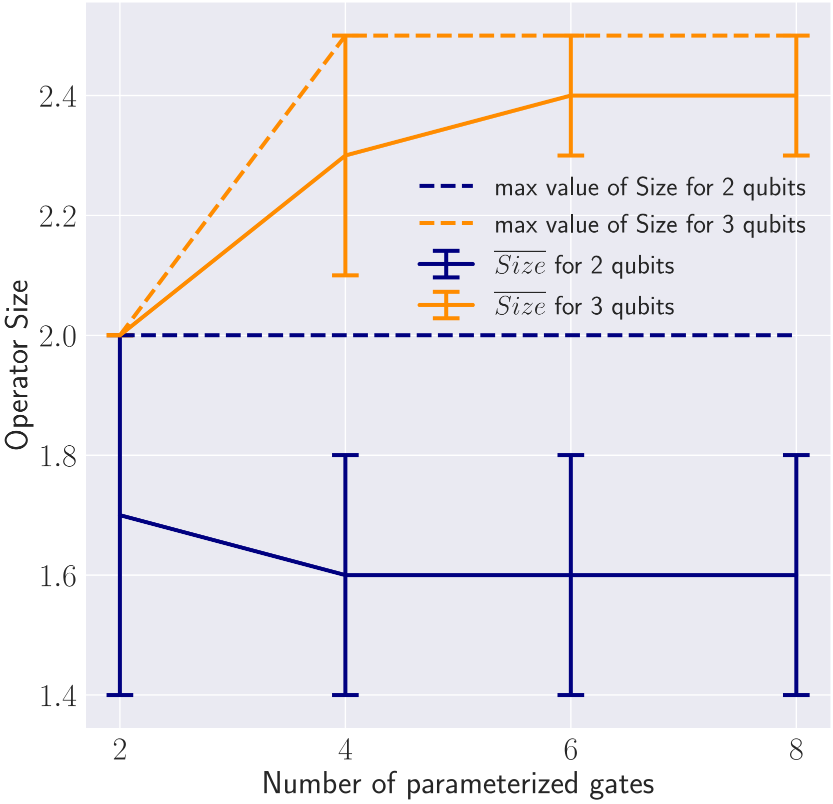


Figure 4.3 We plot the averaged operator sizes defined in Eq.(4.4) and their standard deviation as we increase the number of parameterized and entangling gates. The orange and blue solid lines are for the MC simulation of 2 and 3 qubits, respectively. The dashed orange and blue solid lines give the maximum value that each operator size could get for a given number of parameterized and entangling gates of 2 and 3 qubits, respectively. These results suggest that the 3-qubit circuit architecture is more expressive than the 2-qubit.

the results from the average operator size in the previous section.

We compute the $\overline{\Delta y}$ defined in Eq. 4.5 and see that on average both students are approximately 30% off from the desired target distributions of their teachers. An example of the prediction maps are shown in Fig. D.1 in the Appendix D.2, which suggests that the student learns pretty much uncorrelated labeling. These results are in accordance with the results from the averaged operator size of Fig. 4.3 for two parameterized gates in the sense that both of them have similar values for their averaged operator size, i.e. 2 and approximately 1.7 for the 2-qubit and 3-qubit students, respectively.

Next, we increase the number of parameterized (and entangling gates) by adding a layer L and present the results in Fig. 4.4. We see a significant improvement in performance by increasing the parameterized gates from 2 to 4 for the 3-qubit student. This means that it can learn more reliably the labeling provided by the 2-qubit teacher. On the contrary, the 2-qubit student has almost a constant $\overline{\Delta y}$ for any amount of parameterized gates which suggests that it can't improve and learn more reliably the 3-qubit teacher's outputs.

To obtain these results we used a dataset of 500 points on a 2D grid with $x_i \in [-\pi, \pi]$ on a grid and we generated 100 different labelings via different random initalizations of the teacher. The students are as well randomly initiallized and trained until convergence. These results are in agreement with the calculations from the averaged operator size of the previous subsection 4.3.1. In both cases, for the 3-qubit circuit, we observe an improvement in performance as we increase the number of parameterized gates. Finally, we quickly reach a plateau in performance as we increase further the number of gates as it is also observed in the analysis of the averaged operator size.

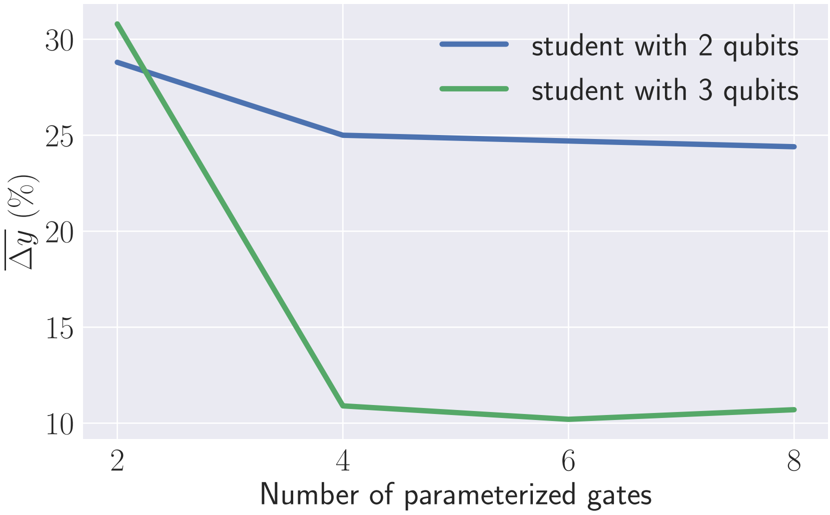


Figure 4.4 We plot the average of the differences between the prediction maps of the teacher and student defined in Eq.(4.4) as we increase the number of parameterized and entangling gates for the simple circuits shown in Fig.4.2. The blue line is for the Teacher with 3 qubits and the student with 2, while the orange line is for the opposite case.

4.3.3 The type of functions that simple quantum models learn

The results from the previous two subsections { 4.3.1 4.3.2} suggest that simple changes in the structure of the circuit highly affects the average operator size and $\overline{\Delta y}$, i.e. the learning capability and expressive power of the models. Therefore, the number of parameterized (and entangling) gates, the circuit architecture and the number of (ancilla) qubits play an important role in QML model's performance. To further analyze the effect of the processing and measurement operators on the expressive power of the models, we use the representation of quantum models as partial Fourier series introduced in subsection 4.2.3.

The Fourier coefficients defined in Eq.4.6 are determined by the eigenvalues of the data-encoding Hamiltonians. For example, the data-encoding Hamiltonian for each qubit from the circuits of Fig. 4.2 is $H = \frac{1}{2}\sigma_x$ and we can assume without loss of generality that it has two distinct eigenvalues, i.e. $\lambda_1 = -1, \lambda_2 = 1$. Importantly, since we have the same type of encoding gate for each qubit, they have the same frequency spectrum Ω. As it is explained in detail in the work of Schuld *et al.* [2020b], we can derive the frequency spectrum for each qubit $\Omega = \{-1, 0, 1\}$ from the possible differences $n_{qubit} = \lambda_j^{qubit} - \lambda_i^{qubit}$ for $\lambda_j^{qubit}, \lambda_i^{qubit} \in \{-1, 1\}$. Then, we have the Fourier coefficients as $c_{n_1 n_2}$ and Eq. (4.6) can be written as

$$f(x) = \sum_{n_1 \in \Omega} \sum_{n_2 \in \Omega} c_{n_1 n_2} e^{-i n_1 x_1} e^{-i n_2 x_2}. \tag{4.7}$$

Both circuit models in Fig. 4.2 have the same number of Fourier basis functions given by Eq. 4.7, since they have the same number of encoding gates. This fundamentally limits their learning ability. But importantly, the structure of the processing and measurement operator affect the distribution of the coefficients.

In Fig. 4.5, we plot the real and imaginary parts of the Fourier coefficients for the 2-qubit (blue color) and 3-qubit (orange color) explicit circuits of Fig. 4.2 with two parameterized gates each. The coefficients $c_{00}, c_{01}, c_{10}, c_{-10}, c_{0,-1}$ from Fig. 4.2 for the 2-qubit circuit are all zero (blue color), while they are non-zero for the 3-qubit (orange color). But what happens with the coefficients $c_{11}, c_{-1,-1}, c_{-11}, c_{1,-1}$? Interestingly, computing the exact analytical formula for the distributions of the coefficient $c_{1,1}$ shows that they are different. We present these analytical formulas in Appendix D.3. In summary, even though the two quantum models of Fig. 4.2 have access to the same Fourier basis functions, the different distribution of their coefficients results in a different combination of these basis functions, and hence, give rise to different function classes.

This is also in accordance with the results from the previous two subsections { 4.3.1 4.3.2}. The 2-qubit student (Fig.4.2 left) has

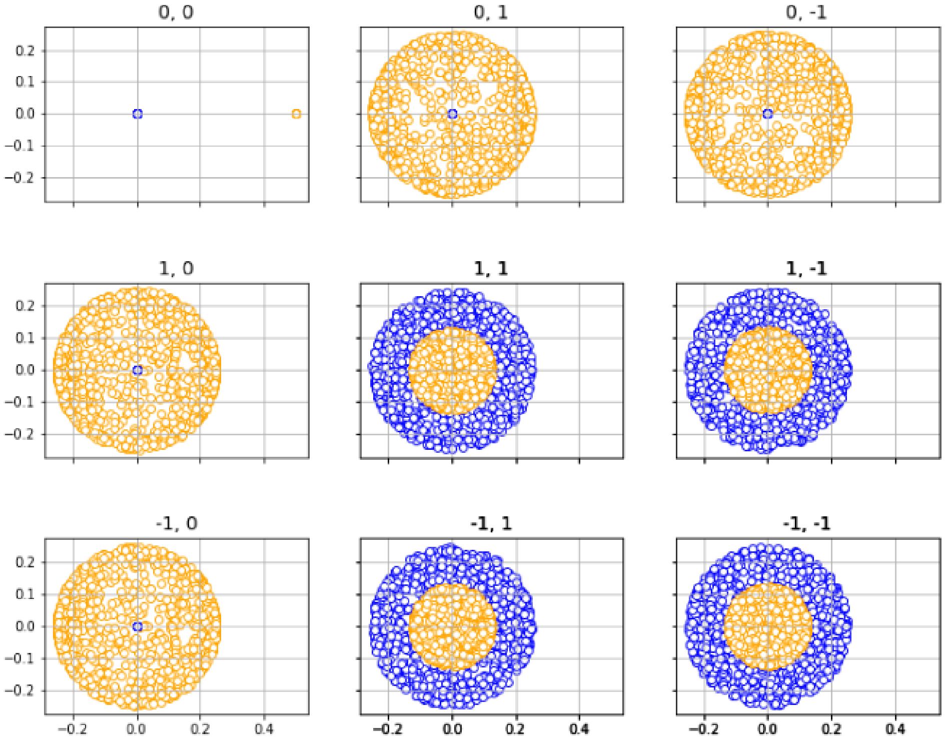


Figure 4.5 The real and imaginary part of the Fourier coefficients for the 2-qubit (blue color) and 3-qubit (orange color) explicit circuits of Fig. 4.2 with two parameterized gates each.

difficulties learning the prediction map of the 3-qubit teacher (Fig.4.2 right), since it has four coefficients strictly equal to zero which does not help approximating the more complicated distribution of the 3-qubit teacher. But reversing the roles, the 3-qubit student has also difficulties learning the 2-qubit teacher. Even though, the student could learn the zero coefficients, the coefficients $c_{11}, c_{-1,-1}, c_{-11}, c_{1,-1}$ of the teacher belong to a different function class, i.e. they draw their values from a different distribution. This could be seen from the larger spread of the coefficients in Fig. 4.5. These results are in accordance with the averaged operator size (Fig.4.3) and $\overline{\Delta y}$ (Fig.4.4), since they also suggest that for just two parameterized gates both the 2-qubit and 3-qubit circuit have similar performance.

Most importantly, once we add an extra layer L to the 3-qubit circuit, the spread of the Fourier coefficients in Figure 4.5 (in orange) increases and overlaps with the coefficients of the 2-qubit circuit (in blue) almost completely. These results are shown in light blue color at Fig.D.3 in the Appendix D.4). This is in accordance with the averaged operator size and $\overline{\Delta y}$ from the teacher-student scheme where a better performance is observed for the 3-qubit case by adding an extra layer L. On the contrary, adding an extra layer in the 2-qubit case does not improve the performance of the student as suggested as well from Fig.4.3 and Fig.4.4, i.e. the distribution of the coefficients shown in Figure 4.5 in blue stays the same.

But what more can we say for the distribution of these coefficients? Could we understand which circuit elements determine the exact zero terms? We tackle these questions in the next section.

4.4 Variational measurement

We can describe the expectation value of any circuit of Fig. 4.1 with the representation of matrix blocks shown in Fig.4.6a. Following Schuld *et al.* [2020b], the data encoding unitary can be written as $S(x) = V^{\dagger}e^{-ix\Sigma}V$ via an eigenvalue decomposition, where Σ is a diagonal matrix of eigenvalues and V is the unitary formed with the eigenvectors as columns. We absorb $V, V^{\dagger}$ into the initial state $|\psi'\rangle = V|\psi\rangle$ and into the processing part of the circuit $U' = VUV^{\dagger}$. For consistency, we transform the measurement operator with $M' = VMV^{\dagger}$. As a result, the encoding unitary is simply a diagonal matrix $\exp(-ix\Sigma)$. The transformed matrix blocks are shown in Fig. 4.6b.

In Fig. 4.7, we schematically illustrate the mapping between the Fourier coefficients and the frequencies instead of explicitly writing the full matrix given by the inner product of the orange, blue and green matrix blocks of Fig. 4.6b. The Fourier coefficients depend only on the processing and measurement operators, therefore the matrix elements of $\overline{M} = U^{\dagger\prime} M' U'$, while the frequencies on the encoding operator, i.e. the diagonal matrix of the encoding Hamiltonian as explained in detail in Schuld *et al.* [2020b]. As introduced in Eq. 4.7, the Fourier coefficients are defined as c_{n_1,n_2}. For example, c_{00} corresponds to the frequencies with $n_1 = n_2 = 0$ for both qubits, which after explicitly calculating the expectation value in Eq. 4.7 in turn correspond to the diagonal matrix elements of $\overline{M}$. Following the same procedure, c_{01} is the sum of elements $\overline{M}_{12}$ and $\overline{M}_{34}$, c_{10} is the sum of elements $\overline{M}_{13}$ and $\overline{M}_{24}$, c_{11} is the element $\overline{M}_{14}$ and $c_{1,-1}$ is the element M_{23}. Finally, $c_{0,-1}$, c_{-10}, c_{-1-1} and c_{-11} are the complex conjugates of c_{01}, c_{10}, c_{11} and c_{1-1}, respectively.

We now discuss the simple example of the 2-qubit case from Fig. 4.2 with the Pauli Z operator applied to the 2nd qubit ($M'_1 = I$, $M'_2 = Z$). The transformed measurement operator $M' = M'_1 \otimes M'_2 = I \otimes \left(VZV^{\dagger}\right)$ of the data-encoding Hamiltonian) becomes

$$M' = \begin{bmatrix} 0 & h & 0 & 0 \\ k & 0 & 0 & 0 \\ 0 & 0 & 0 & h^* \\ 0 & 0 & k^* & 0 \end{bmatrix}, \tag{4.8}$$

where h, k are complex trigonometric functions that determine the distribution of the coefficients. But the mapping between the zero elements and the coefficients that are zero does not match. For example, c_{11} which is determined by $\overline{M}_{14}$ has a non zero distribution as shown in Fig.4.5 (blue color), while eq. (4.8) suggests that $\overline{M}_{14}$ it should be strictly zero.

To resolve this discrepancy, we need to consider the processing and measurement operators as one matrix block $\overline{M} = U^{\dagger\prime} M' U'$ (shown in green in Fig. 4.6c). This is in accordance with recent works Schuld [2021a]; Jerbi *et al.* [2021], where the processing and measurement operators are regarded as one part in the circuit structure.

For the 2-qubit case, the combined matrix $\overline{M}$ (green blocks in Fig.4.6c) becomes:

$$\overline{M} = \begin{bmatrix} 0 & 0 & 0 & f \\ 0 & 0 & g & 0 \\ 0 & g* & 0 & 0 \\ f* & 0 & 0 & 0 \end{bmatrix}, \tag{4.9}$$

where f, g are complex trigonometric functions that determine the distribution of the coefficients shown in Fig. 4.5. Following the mapping between the coefficients and the elements of the combined matrix $\overline{M}$, we can immediately see which coefficients are zero. The only non-zero coefficients are the c_{11}, $c_{1,-1}$ and their complex conjugates determined by $\overline{M}_{14}$, $\overline{M}_{23}$ and $\overline{M}_{41}$,$\overline{M}_{32}$, respectively.

This analysis suggests that in some cases replacing the usual segregation of a quantum circuit into three parts: encoding, processing and measurement with just two parts: encoding and variational measurement could reveal further insights on the performance and properties of QML models.

4.4.1 Toy application

To further emphasize the importance of the measurement operators of a circuit, we present a simple quantum model with two qubits (Fig.4.8). The Fourier coefficients of this circuit are represented in Fig.4.9, where the coefficients in purple represent the measurement of the 1st qubit and the Fourier coefficients in green represent the measurement on the 2nd qubit. As we see, the distribution of the Fourier coefficients differ greatly. Therefore, with the same circuit structure, we could create two different functions classes depending on where we measure. This might have a more relevant applications, but it already suggests that once we better understand how simple elements on our circuit structure affect the overall performance it is more natural to think of applications.

4.5 Discussion

The results presented in this chapter gives us a better understanding on the basic questions posed earlier in the introduction: "How many parameterized gates to use?", "How many entangling gates and where?", "Do ancilla qubits help?" and "Where to measure?". If we want to find the great applications that QML promises Alcazar and Perdomo-Ortiz [2021]; Schuld and Killoran [2022b], we first need to understand how QML models work. The data encoding fundamentally limits the performance of a circuit, but as we also showed in this chapter the processing and measurement also significantly affect the capabilities of the model. We showed that different parts of processing and measurement operators affect the performance of the QML model.

To quantify the model's performance, we introduced three different tools: the averaged operator size, the $\overline{\Delta_y}$ and the coefficient from the partial Fourier series. By applying the Fourier representation, we observed that two different circuits with the same number of parameters could give rise to different function classes. Finally, we found that the segregation of the processing and measurement operator might not be ideal when interpreting the parts of the circuit. Instead when considered as one, i.e. the variational measurement, could give insights on the model's performance. The idea of the variational measurement has also been presented in the context of quantum kernels Schuld [2021b]. This might bring the relation between the Fourier representation and quantum kernels one step closer.

Once we better understand how QML models work, we can naturally start thinking about applications. For example, we presented a trivial implementation that arises from where we measure. Specifically, measuring either one of the two qubits on a 2-qubit circuit gives rise to fundamentally different classes of functions that this model could express. Therefore, with the same circuit structure one could learn or express two different function classes as shown by the numerical calculation of the real and imaginary part of the Fourier coefficients in Fig. 4.8. We understand that computing the analytical trigonometric expression for more qubits, or even gates, is similar to opening Pandora's box. Instead we suggest to focus on the other two tools proposed, i.e. the averaged operator size and the map differencies from the Teacher-student scheme.

In this work, we focused on simple quantum models and verified that the tools presented here are robust. The analytical and numerical results are in agreement, and therefore, these tools could be further used to test the performance of more complex circuit structures. Moreover, applying the averaged operator size and Teacher-student scheme to other simple circuit structures could further help understand how QML models work. It would also be interesting to exploit the effect of measurement for a more interesting use-case application. But to do so, a careful study on how the measurement position affects the model's performance should be undertaken, i.e. by increasing the number of qubits and circuit complexity Haferkamp *et al.* [2022]. This study could also further strengthen the belief that quantum models with ancillary qubits have better performance overall Du *et al.* [2020].

4.6 Conclusion

In this work we explored the performance of small quantum models from different perspectives, i.e. their training capability and information scrambling. This study helps us better understand how different elements of the quantum circuits affect its overall performance. Moving forward these tools could potentially be applied to larger scale quantum simulations.

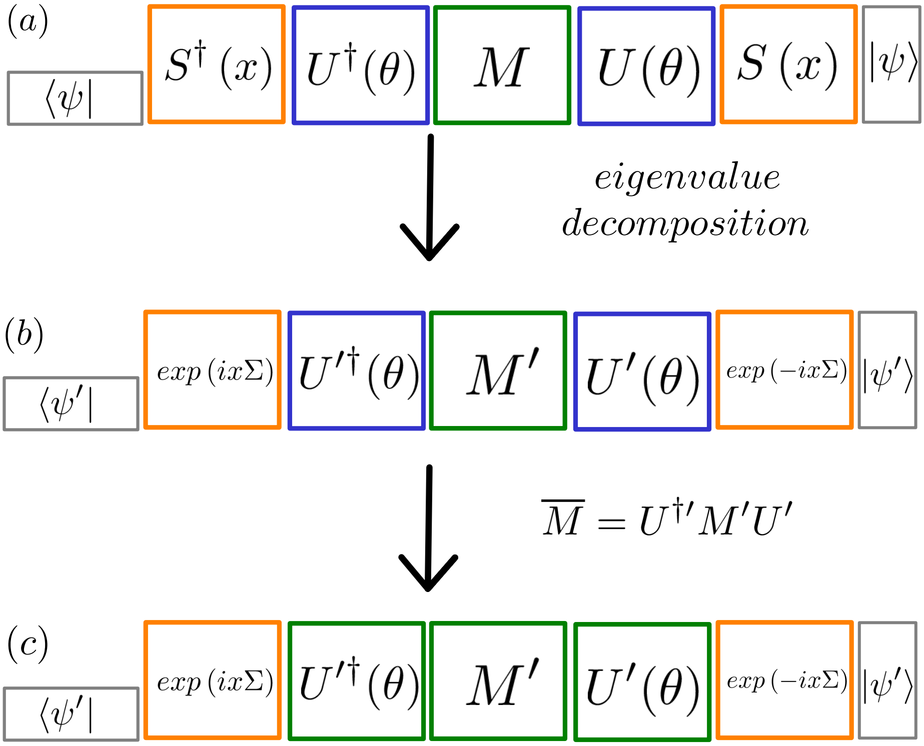
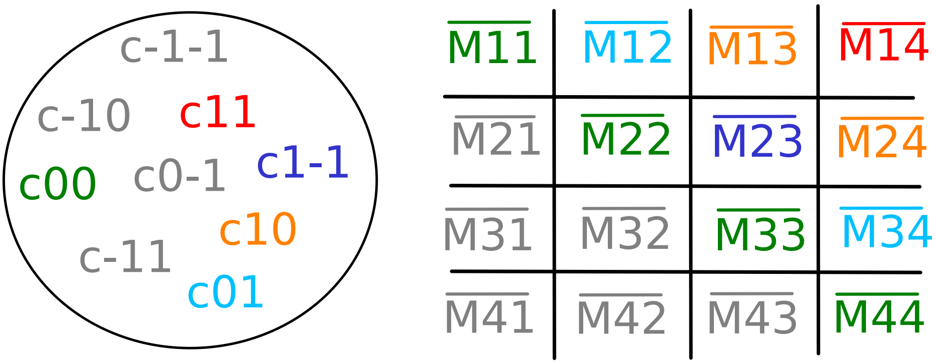


Figure 4.6 We illustrate the matrix block representation of the expectation value of any circuit given by Fig. 4.1 (a), after following the Fourier analysis, i.e. with the eigenvalue decomposition of the encoding unitaries (orange color) (b) and after considering as one the processing and measurement operators shown in green color (c).



$\overline{M11}$	$\overline{M12}$	$\overline{M13}$	$\overline{M14}$
$\overline{M21}$	$\overline{M22}$	$\overline{M23}$	$\overline{M24}$
$\overline{M31}$	$\overline{M32}$	$\overline{M33}$	$\overline{M34}$
$\overline{M41}$	$\overline{M42}$	$\overline{M43}$	$\overline{M44}$

Figure 4.7 The mapping between the Fourier coefficient and the matrix elements of an arbitrary matrix $\overline{M}$. The colors represent the five independent Fourier coefficients from the nine coefficients in total, if we exclude the complex conjugates shown in grey color.

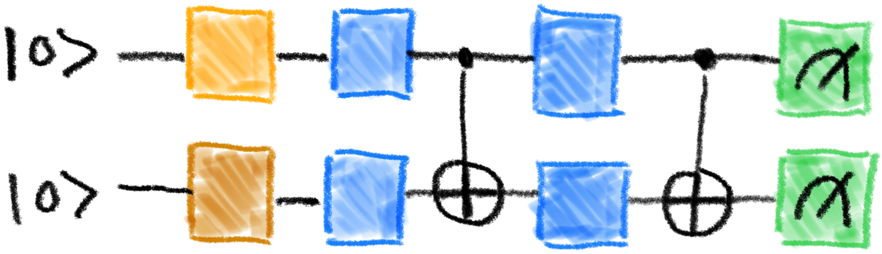


Figure 4.8 A 2-qubit circuit with 4 parameterized and two entangling gates where we could either measure on the 1st or 2nd qubit.

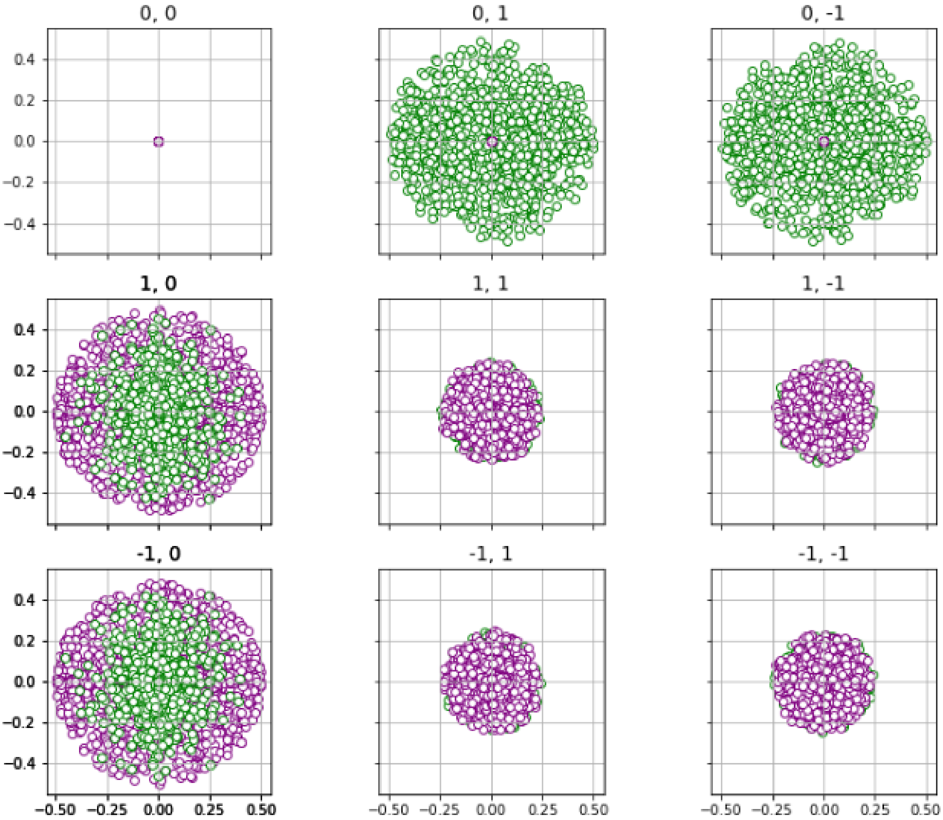


Figure 4.9 The real and imaginary part of the Fourier coefficients of the circuit structure shown in Fig. 4.8. If the circuit is measure on the first qubit we obtain the Fourier coefficients in purple, if we measure on the second qubit we obtain the coefficients in green.

Chapter 5

Benchmarking quantum algorithms for quantum chemistry

In the Section 4.4.1 of the previous chapter, we got a glimpse at the potential of quantum algorithms, and specifically, variational quantum algorithms for realizing toy applications. In this chapter, we shift our focus to the field of quantum chemistry which is one of the main industrial relevant fields that promise to be the killer application of quantum computation. Specifically, quantum computation promises to unlock new computational capabilities for certain tasks such as the ground state energy estimation (GSEE) for molecules and materials Aspuru-Guzik *et al.* [2005]; Goings *et al.* [2022b]; Elfving *et al.* [2020a].

However, realizing quantum advantage for the task of GSEE requires improvements in the quantum algorithms that will reduce the resource requirements needed, such as the circuit depth Wang *et al.* [2022d]. This has led to extensive research on the development of algorithms with modest circuit depths Wang *et al.* [2022d]; Ding and Lin [2022a]. But the performance of GSEE algorithms also strongly depends on the overlap of the true ground state of the Hamiltonian and the initial state generated by a ground state preparation (GSP) method Zhang *et al.* [2022]; Dong *et al.* [2022a]; Lee *et al.* [2022]. For quantum chemistry applications, the Hartree-Fock (HF) Slater determinant state is widely used for GSP, since the cost (in terms of circuit depth) of implementing it on quantum hardware is insignificant compared to GSEE algorithms Nielsen and Chuang [2011] and it provides satisfactory results for many molecules and materials Tubman *et al.* [2018]. However, in some important cases the overlap is relatively small (for example molecules with a bond distance out of equilibrium Tubman *et al.* [2018]; Ceroni *et al.* [2022]), which creates a need for methods that can provide a larger initial overlap.

The notion of "good overlap", as usually referred to, is vague and does not explore a performance to resource cost ratio as a benchmarking tool Tubman *et al.* [2018]; Lee *et al.* [2022]. In this chapter, we introduce the *acceptability criteria* that address the trade-off between the resource cost and performance improvement of the GSP and GSEE subroutines. Such a tool could be used to answer questions like: Is it worth the high circuit depth cost to use a GSP algorithm that provides almost perfect overlap values? Or is it better to settle for a heuristic method like VQE with smaller ground state overlap even though it increases the runtime of the GSEE subroutine? It provides a reliable tool to benchmark the performance of the plethora of GSP algorithms for the task of GSEE.

The original results presented in this chapter are based on Gratsea *et al.* [2024], a joint collaboration with C. Sun and P. D. Johnson.

5.1 Evaluating the efficiency of Ground State Preparation (GSP) algorithms

5.1.1 Acceptability criteria of GSP algorithms

The *efficiency* of a GSP algorithm gives the right tools to understand the appropriate balance of resource cost and performance of GSP and GSEE algorithms. To evaluate the performance of quantum algorithms, recent work proposed a resource efficiency metric as the ratio of the success metric over the resource cost Auffè ves [2022]. Instead of defining efficiency metrics, in this work we introduce a criteria to evaluate whether to accept or reject a given GSP algorithm. Specifically, we propose a systematic way to benchmark GSP methods for the problem of GSEE. We use the HF method as a reference and explore under which conditions a GSP method will be accepted over HF. The benchmarking criteria incorporate both the reduction of the total runtime for GSEE and the resource cost of the GSP algorithm (see Fig.1). We perform numerical simulations to showcase how to use the criteria in practice and provide a resource estimation of the maximum allowed depth of a GSP to be acceptable over HF.

The chapter is organized as follows: in Sec. 5.1.2 we introduce the criteria for acceptability of GSP that do not require repetitions, while we incorporate repetitions of GSP in Sec. 5.1.3. In Sec. 5.1.4, we present numerical simulations on how to use the benchmarking criteria set in the first two sections. Also, we include a resource estimation of the maximum allowed GSP circuit depth for solid-state materials. Sec. 5.1.5, contains the conclusions and future research directions.

5.1.2 Criteria for acceptability of state preparation without repetitions

Here we discuss the criteria under which a state preparation method is acceptable for the purposes of GSEE. Any ground state energy estimation method has a runtime that depends on the features of the ground state preparation: the GSP circuit depth D and the overlap between the prepared state and the ground state $\eta = \gamma^2$. The runtime also depends on the target accuracy ϵ. For a given energy estimation

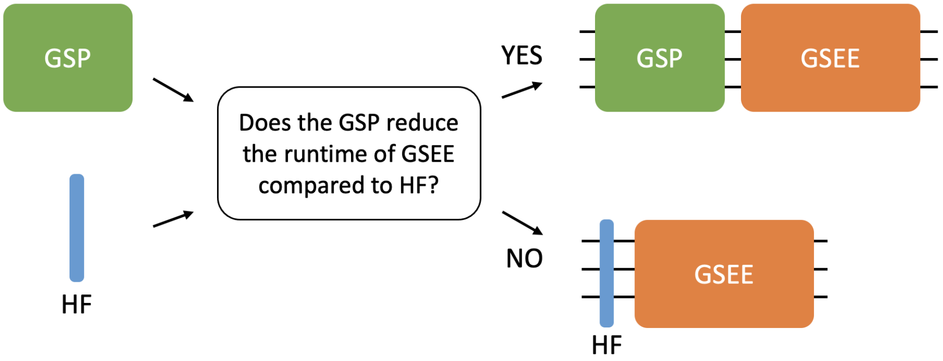


Figure 5.1 The acceptability criteria is used to benchmark the given GSP method over the HF reference with the goal of reducing the total runtime of the GSEE algorithm.

algorithm Dong *et al.* [2022a], the runtime can be formulated as

$$\begin{aligned}\mathcal{T}(\mathcal{D},\gamma;\epsilon) &= (\text{number of repetitions}) \\ &\times(\text{total circuit depth of each repetition}) \\ &= (\text{number of repetitions}) \\ &\times(\text{GSP depth} + \text{GSEE depth}) \\ &= \tilde{O}\left(\frac{1}{\gamma^{\alpha}}\left(D+\frac{1}{\epsilon\gamma^{\beta}}\right)\right). \end{aligned} \tag{5.1}$$

According to the table given in reference Dong *et al.* [2022a], example GSEE algorithms have values $\alpha \in \{0,2,4\}$ and $\beta \in \{0,1,2\}$. The units for the GSP and GSEE circuit depths need to match, but as it will become evident from the numerical simulations in the next section different choices for the units could be used, such as the circuit depth and the T-gate count. Also, we are ignoring constant factors and logarithmic dependence on the parameters for now to simplify the introduction of this technique; but these should ultimately be included to set a more accurate benchmark. Finally, some state preparation methods require a number of repetitions to ensure their success (with high probability). In Sec. 5.1.3, we discuss the runtime cost that includes repetitions of the GSP algorithms.

To establish the concept of an *acceptable* state preparation, we must assume that the ground state energy estimation algorithm has a reference or default method to compare with. A reference initial state could be simply a product state of the measurement basis, or the solution to a mean-field-level method, such as the Hartree-Fock ground state. The circuit depth for preparing a reference state can usually be neglected, thus we label the depth and overlap as $D_0 = 0$ and γ_0. In this work, we focus on the HF method as a reference exactly due to the zero depth cost. Other methods, such as adiabatic state preparation Aspuru-Guzik *et al.* [2005]; Albash and Lidar [2018], which have non-zero depth cost, could be used as reference in future work. The runtime of GSEE using the HF reference state preparation is

$$\mathcal{T}_0(D_0, \gamma_0; \epsilon) = \tilde{O}\left(\frac{1}{\epsilon\gamma_0^{\alpha+\beta}}\right). \tag{5.2}$$

The condition for a state preparation method to be acceptable over the reference is that the total runtime of the GSEE with the GSP ($\mathcal{T}$) is smaller than the total runtime of the GSEE with HF ($\mathcal{T}_0$), i.e. $\mathcal{T} < \mathcal{T}_0$. This puts constraints on the state preparation parameters:

$$\frac{1}{\gamma^\alpha}\left(D + \frac{1}{\epsilon\gamma^\beta}\right) < \frac{1}{\epsilon\gamma_0^{\alpha+\beta}}. \tag{5.3}$$

Observe that if $D = 0$, then the acceptability criteria reduce to the condition $\gamma > \gamma_0$. We can rewrite the general condition (5.3) as

$$D < \frac{1}{\epsilon\gamma^\beta}\left(\left(\frac{\gamma}{\gamma_0}\right)^{\alpha+\beta} - 1\right). \tag{5.4}$$

This shows that if ϵ is decreased then a state preparation with larger D will be accepted. In other words, for less demanding GSEE algorithms with a worse target accuracy ϵ more costly GSP algorithm could be accepted over the HF. Finally, we could write the above inequality as following

$$\frac{D + 1/\epsilon\gamma^{\beta}}{1/\epsilon\gamma^{\beta}} < \left(\frac{\gamma}{\gamma_0}\right)^{\alpha+\beta} . \tag{5.5}$$

Therefore, the acceptability criterion in the more strict case when $\alpha + \beta = 1$, could be expressed in words as

$$\frac{\text{total depth}}{\text{GSEE depth}} < \frac{\text{gsp overlap}}{\text{HF overlap}} \tag{5.6}$$

or

$$\frac{\text{total depth}}{\text{GSEE depth}} < \frac{N_{reps} \text{ from HF}}{N_{0,reps} \text{ from GSP}}, \tag{5.7}$$

where N_{reps} is the number of repetitions due to the GSP overlap value, i.e. $N_{reps} = 1/\gamma$ and $N_{0,reps} = 1/\gamma_0$.

Next, we discuss the simple case when the GSP query depth is much smaller than the GSEE query depth. Then, we have

$$\frac{\text{GSP depth}}{\text{GSEE depth}} \ll 1 \tag{5.8}$$

$$D\gamma^{\beta} \ll \frac{1}{\epsilon}. \tag{5.9}$$

For typical values of $\epsilon \simeq 10^{-3}$ Wang *et al.* [2022d]; Babbush *et al.* [2018a], the condition becomes $D\gamma^{\beta} \ll 10^3$. Since $\beta \in \{0, 1, 2\}$ Dong *et al.* [2022a] and $\gamma \leq 1$, the more strict condition is $D \ll \frac{1}{\epsilon}$.

Then, the acceptability criteria for any $\alpha \in \{0, 2, 4\}, \beta \in \{0, 1, 2\}$ Dong *et al.* [2022a] is simplified to

$$1 < \left(\frac{\gamma}{\gamma_0}\right)^{\alpha+\beta}, \tag{5.10}$$

which simply states that the acceptance of the GSP over HF is determined by the respective overlap values ratio.

Next, we compare the acceptability criteria for two different GSEE algorithms presented in the table of GSEE performance Dong *et al.* [2022a], the quantum phase estimation semi-classical (QPE) Berry *et al.* [2009, 2015] and the GSEE algorithm developed by Tong et. al. (referred as LT20) Dong *et al.* [2022a] for a given GSP algorithm introduced in Lin and Tong [2020a] whose depth depends on the lower bound of the spectral gap Δ and the overlap γ_0. For QPE we have $\alpha = \beta = 2$ and

$$\frac{\epsilon\gamma^2 + \Delta\gamma_0}{\Delta\gamma_0} < \left(\frac{\gamma}{\gamma_0}\right)^4, \tag{5.11}$$

while for LT20 we have $\alpha = 0, \beta = 1$ and

$$\frac{\epsilon\gamma + \Delta\gamma_0}{\Delta\gamma_0} < \left(\frac{\gamma}{\gamma_0}\right). \tag{5.12}$$

Since $\gamma \leq 1$, we have

$$\frac{\epsilon\gamma^2 + \Delta\gamma_0}{\Delta\gamma_0} \leq \frac{\epsilon\gamma + \Delta\gamma_0}{\Delta\gamma_0} < \left(\frac{\gamma}{\gamma_0}\right) < \left(\frac{\gamma}{\gamma_0}\right)^4. \tag{5.13}$$

This suggests that the acceptability criteria for the LT20 is more strict than the QPE. The LT20 has a smaller GSEE query depth compared to QPE, so it is harder to accept a GSP algorithm with non-zero depth over the HF. Therefore, the better the GSEE algorithms becomes in terms of query depth reduction, the more strict the criteria for the acceptance of a GSP method over HF. In other words, as the query depth of the GSEE becomes smaller, the number of repetitions imposed by the overlap prepared from the GSP algorithm becomes less significant.

Finally, the acceptability criteria allow us to explore the maximum values of the GSP depth that enable the given GSP method to be acceptable over the HF state. Given a specific GSEE algorithm and the value of γ_0, and assuming that the GSP provides a specific value of γ, i.e. $\gamma = 1$, we find the corresponding maximum acceptable depth of a GSP method. To this end, Eq. (5.5) can be written as

$$D < \frac{\gamma - \gamma_0}{\gamma_0} D_{\text{GSEE}}, \tag{5.14}$$

for more demanding GSEE algorithms with $\alpha + \beta = 1$ and depth D_{GSEE}. The above equation can be expressed as

$$\text{GSP depth} < \frac{\text{performance gain}}{\text{HF performance}} \text{GSEE depth} . \tag{5.15}$$

5.1.3 Criteria for acceptability of state preparation methods with repetitions

In this section, we discuss the criteria under which a GSP method that requires repetitions to reach an overlap γ is acceptable for a given GSEE algorithm. The runtime Eq. (5.1) discussed in the previous section becomes

$$\begin{aligned} \mathcal{T}(\mathcal{D}, \gamma; \epsilon) = & \text{(number of repetitions of GSEE)} \\ & \times[\text{(number of repetitions of GSP)} \\ & \times\text{(circuit depth of GSP)} \\ & +\text{(circuit depth of GSEE)}] \\ = & \tilde{O}\left(\frac{1}{\gamma^{\alpha}}\left(\frac{1}{P_{\text{succ}}} \times D + \frac{1}{\epsilon\gamma^{\beta}}\right)\right). \end{aligned} \tag{5.16}$$

As explained earlier, the runtime for the GSEE using the HF reference state preparation $\mathcal{T}_0$ is given by Eq. (5.2). Then, the condition, i.e. $\mathcal{T} < \mathcal{T}_0$, for a state preparation method being acceptable over the reference becomes

$$\frac{D/P_{succ} + 1/\epsilon\gamma^{\beta}}{1/\epsilon\gamma^{\beta}} < \left(\frac{\gamma}{\gamma_0}\right)^{\alpha+\beta}, \tag{5.17}$$

which can be expressed as

$$\frac{\text{total query depth}}{\text{GSEE query depth}} < \frac{N_{\text{reps}} \text{ from HF}}{N_{reps} \text{ from GSP}}.$$

5.1.4 Preliminary numerical simulations

In this section, we apply the acceptability criteria and benchmark different GSP methods over the HF for different Hamiltonians of molecules and solid-state materials, starting from small molecules (H_2 molecule) and moving on to larger molecules (N_2 molecule). Finally, we perform a resource estimation of the maximal acceptable circuit depth of GSP for different solid-state materials over HF state. For the numerical simulations we followed the notebook Kottmann and Aspuru-Guzik [2022b] and the molecular data Kottmann *et al.* [2021b].

Molecules We explore the acceptability criteria for a small molecule (H_2) with 4 spin-orbitals or qubits in an adapted basis Kottmann *et al.* [2021b]. We compare the HF method to the separable pair approximation (SPA) approach introduced in the recent work of Kottmann and Aspuru-Guzik [2022b] as the GSP method. According to this work, the circuit depth of SPA for the H_2 molecule equals to 3. Since the depth is $D_{GSP} = 3 << 10^3$ for typical values of chemical accuracy $\epsilon = 10^{-3}$, we are in the simple case discussed in Sec. 5.1.2 where the criteria are simplified to the overlap values ratio (Eq. (5.10)). We assume a more demanding GSEE algorithms with $\alpha + \beta = 1$ and the criteria is given by

$$1 < \frac{\gamma}{\gamma_0}. \tag{5.18}$$

In Fig.5.2, we plot the fidelity of the two different GSP methods. For all bond distances presented in Fig.5.2, the criteria is satisfied. Specifically, at bond distance $d = 0.5$ the ratio $\frac{\gamma}{\gamma_0}$ is 1.005 leading up to the value of 1.5 for $d = 2.6$. This suggests that initially SPA is comparable to HF and as we increase the bond distance (Å), SPA is acceptable over HF. For a less demanding GSEE algorithm (i.e. with $\alpha + \beta$ possessing different values than 1), the criteria would be satisfied and SPA would be acceptable over the HF method.

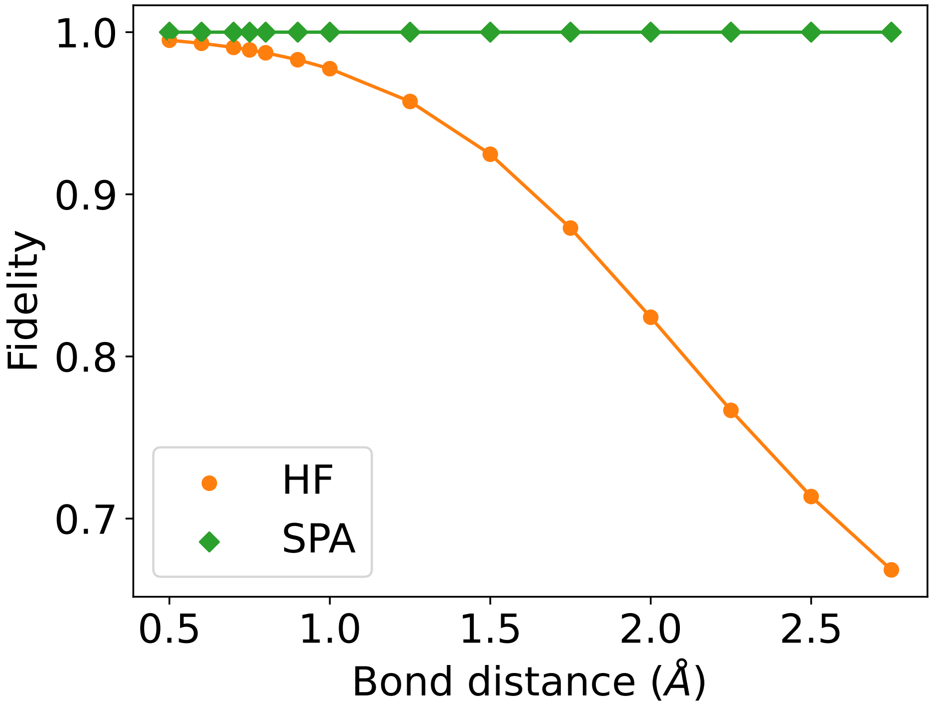


Figure 5.2 Fidelity as a function of bond distance of the H_2 molecule for the HF method and the SPA algorithm as GSP methods respectively. The value of alpha and beta is set to be alpha + beta = 1, which corresponds to a GSEE method that has a more strict acceptability criteria.

Next, we explore the larger molecule N_2 considering 6 active electrons in 12 spin-orbitals or qubits in an adapted basis Kottmann *et al.* [2021b] at bond distance $d = 2.0$. Initially, we benchmark the SPA method over the HF. Since $D_{GSP} = 3$ Kottmann and Aspuru-

Guzik [2022b], we could apply the criteria of Eq.5.18 which gives

$$1 < \frac{\gamma}{\gamma_0} = \frac{0.85}{0.72} = 1.18. \tag{5.19}$$

Therefore, the SPA is acceptable over the HF method.

Next, we benchmark a more costly heuristic algorithm: the low-depth-booster from the recent work of Wang *et al.* [2022b]. To this end, we change the unit of depth from the circuit depth to the accumulations of the controlled time evolution $epx\,(2i\pi H)$ operations, where H is the Hamiltonian of the system. Following the aforementioned work Wang *et al.* [2022b], we have $D_{GSP} = 10^3$ with $\gamma \approx 1$, while $D_{GSEE} = 2 \times 10^4$ and $\gamma_0 = 0.72$. The success probability of the low-depth-booster GSP algorithm applied with the linear combination of unitaries (LCU) method is $P_{succ} \approx 0.5$ Wang *et al.* [2022b]. Therefore, the criteria of Eq. 5.17 becomes

$$\frac{D/(P_{succ}) + 1/\epsilon\gamma^{\beta}}{1/\epsilon\gamma^{\beta}} < \left(\frac{\gamma}{\gamma_0}\right)^{\alpha+\beta} \Rightarrow \tag{5.20}$$

$$\frac{2.2}{2} < \left(\frac{1}{0.72}\right)^{\alpha+\beta} \Rightarrow 1.1 < (1.39)^{\alpha+\beta}\,, \tag{5.21}$$

which is satisfied for any values of α, β of the GSEE algorithms.

Solid-state materials Here, we perform a resource estimation of the maximal acceptable depth of the GSP methods for different solid-state materials.

Recent work Babbush *et al.* [2018a] estimates the T gates needed for quantum simulation of 3D spinful jellium (or the homogeneous electron gas). It focuses on T-count since applying a T gate requires a lot of logical qubits and takes much longer than any other operation in a quantum circuit Fowler and Devitt [2012]. The 3D spinful jellium is in the dual basis at Wigner-Seitz radius of 10 Bohr radii assuming the system is at half filling. For 54 spin-orbitals and a target chemical accuracy $\Delta E = 0.0016$ Hartree the depth is equal to 1.8×10^7 T-count.

As explained in the work of Babbush *et al.* [2018a], the jellium is a good proxy for different solid-state materials, such as diamond, graphite, silicon, metallic lithium and crystalline lithium hydride. For these materials, the HF overlap could range from smaller to larger values as presented in Fig. 5.3. Assuming that the GSP method gives $\gamma \approx 1$, we have a resource estimation of the maximum depth allowed for the GSP method to be acceptable over HF (see Fig. 5.3) given by Eq. (5.14).

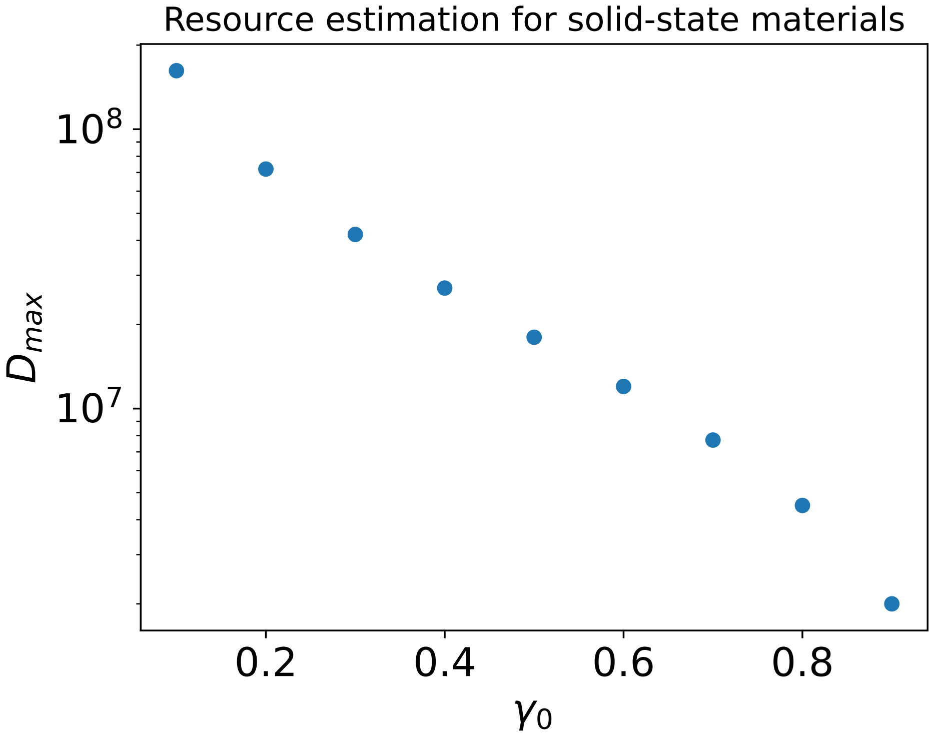


Figure 5.3 Maximum acceptable depth D_{max} of the GSP algorithm with $\gamma = 1$ and $D_{GSEE} = 1.8 \times 10^7$ for solid state materials as a function of the HF overlap γ_0.

5.1.5 Discussion

We introduced a method to assess when to accept or reject a ground state preparation (GSP) method over the Hartree-Fock (HF) reference for the task of ground state energy estimation (GSEE) by introducing acceptability criteria. The criteria are defined through the total runtime of the GSEE algorithm that incorporates both the number of repetitions needed and the total circuit depth of each repetition—i.e. the GSP and the GSEE depth. If the inequality introduced in Eq. 5.5 is satisfied, then the GSP method is acceptable over the HF—i.e. provides a speedup in the total runtime of the GSEE algorithm. The criteria explores the trade-off of both the resource cost and performance of GSP and GSEE subroutines.

We explored under which conditions the acceptability criteria could be simplified and also established them for GSP methods that require repetitions to reach an overlap γ. Comparing the acceptability criteria for two different GSEE algorithms with a GSP, we found that the better the GSEE algorithms becomes in terms of query depth reduction, the more strict the criteria to accept a GSP over the HF. This could be due to the fact that the number of repetitions introduced by the GSP overlap becomes less significant as the GSEE query depth becomes smaller. The ability to trade circuit depth with runtime is also motivated in recent works Wang *et al.* [2022d]; Ding and Lin [2022a]. In agreement, the resource estimation performed in this work suggests that a GSP method with a larger circuit depth than the GSEE could be accepted for total runtime reduction.

Next, we showed that the separable pair approximation (SPA) method is acceptable over the HF for the hydrogen molecule for different bond lengths, which suggests that even for a simple molecule, there exists GSP that could offer an improved performance to the GSEE algorithm over using HF. We also evaluated the more expensive low-depth-booster GSP algorithm for the nitrogen molecule which is widely used for benchmarking quantum chemistry simulations; in particular, when the bond is stretched Tubman *et al.* [2018]; Kottmann and Aspuru-Guzik [2022b]; Wang *et al.* [2022b]. These results suggest that more expensive GSP methods could reduce the total runtime, thus being acceptable over the HF reference. In accordance, the resource estimation of the maximum allowed depth of a GSP does not provide evidence against the use of VQE and more expensive heuristic methods. Further numerical and theoretical work is needed to draw a more definitive conclusion.

This work sets a foundation to further explore resource efficiency metrics for GSP and GSEE algorithms. It would be interesting to apply the criteria introduced here to molecules and materials of industrial relevance Gonthier *et al.* [2022a] and further use them for resource estimations. Moreover, they could be adjusted to incorporate logarithmic dependencies on the parameters or the recent GSEE algorithm with an exponential improvement in the circuit depth Wang *et al.* [2022d]. Finally, other GSP methods could play the role of the reference method instead of the HF. It is challenging to assess what combination of methods will ultimately be used in practice, and further research will help evaluate the utility of various combinations of GSEE and GSP methods.

5.2 Comparing Classical and Quantum Ground State Preparation Heuristics

As already discussed in the previous section, one promising field of quantum computation is the simulation of quantum systems, and specifically, the task of ground state energy estimation (GSEE). Ground state preparation (GSP) is a crucial component in GSEE algorithms, and classical methods like Hartree-Fock state preparation are commonly used. However, the efficiency of such classical methods diminishes exponentially with increasing system size in certain cases. In this study, we investigated whether in those cases quantum heuristic GSP methods could improve the overlap values compared to Hartree-Fock. Moreover, we carefully studied the performance gain for GSEE algorithms by exploring the trade-off between the overlap improvement and the associated resource cost in terms of T-gates of the GSP algorithm. Our findings indicate that quantum heuristic GSP can accelerate GSEE tasks, already for computationally affordable strongly-correlated systems of intermediate size. These results suggest that quantum heuristic GSP has the potential to significantly reduce the runtime requirements of GSEE algorithms, thereby enhancing their suitability for implementation on quantum hardware.

5.2.1 Importance of numerical simulations and resource estimations

The simulation of quantum systems is one of the more promising applications of quantum computers Cao *et al.* [2019]. In particular, many methods have been proposed to solve the ubiquitous task of ground state energy estimation Poulin *et al.* [2018]; Lin and Tong [2022a]; Wang *et al.* [2022d]. Unfortunately, the quantum resources needed to solve this task for industrial applications is many millions of physical qubits Kim *et al.* [2022a]; Goings *et al.* [2022c] and the computations can take days to months to run. The value of large-scale quantum computing will depend on the degree to which these costs can be reduced Lee *et al.* [2022]; Katabarwa *et al.* [2023].

For the task of ground state energy estimation in both classical and quantum computing, a critical subtask is *ground state preparation*. Examples of approximate ground state preparation include Hartree-Fock (HF) Echenique and Alonso [2007], configuration interaction (CI) Sherrill and Schaefer III [1999], and density-matrix renormalization (DMRG) Wouters *et al.* [2014]; Zhai *et al.* [2023]. The performance of the ground state energy estimation method depends on the quality of an approximate ground state, known as a trial or ansatz state Tubman *et al.* [2018]; Zhang *et al.* [2022]; Ding and Lin [2022a]; Dong *et al.* [2022a]; the greater the magnitude of the inner product or *overlap* between the ansatz and the true ground state, the more efficient the ground state energy estimation can be.

This dependence has led to an increased interest in approximate ground state preparation (GSP) algorithms Tubman *et al.* [2018]; Fomichev *et al.* [2023]. The Hartree-Fock state preparation, one of the most established classical methods for state preparation, is often the baseline method for ground state preparation in quantum algorithms for quantum chemistry. HF state preparation works well for many systems Tubman *et al.* [2018], but in strongly correlated cases, the overlap can decrease exponentially with the system size Lee *et al.* [2022]. Recent works Gratsea *et al.* [2022]; Pathak *et al.* [2023] discuss the cost-benefit ratio of quantum GSP over HF for the task of GSEE, while other works focus on more expensive and accurate classical GSP methods Elfving *et al.* [2020b]; Lee *et al.* [2022]; Babbush *et al.* [2018b]. If quantum GSP methods offer a better trade-off between performance and resource cost over HF, then the same holds for most other classical chemistry methods Babbush *et al.* [2023].

Regardless of the recent progress in the field, there is still ongoing research on how quantum GSP methods could significantly improve the task of GSEE. A lot of progress has been made in GSP algorithmic development, but it remains unclear how costly these quantum heuristic algorithms are in regard to GSEE algorithms. To tackle such questions, numerical simulations of quantum heuristic GSP are necessary to reach efficient ground state preparation as emphasized in Lee *et al.* [2022]. In this work, we carefully evaluate the trade-off of the resource cost in terms of the gate count for an improved overlap value of the studied quantum heuristic GSP over classical methods Gratsea *et al.* [2022].

Towards this goal, we perform numerical simulations and resource estimations motivated by the following questions:

- To what degree can GSP algorithms improve the ground state overlap when the corresponding HF overlap has small values? (see Sec. 5.2.2 and Fig. 5.4)

- Could quantum heuristic algorithms continue to improve the ground state overlap beyond that of HF as we increase the system size? (see Sec. 5.2.3 and Fig. 5.5)

- Assuming the quantum GSP algorithms as subroutines of GSEE, to what degree do they improve the performance while maintaining moderate computational cost? (see Sec. 5.2.4 and Fig. 5.6)

We focus on two *heuristic* quantum GSP methods, the variational quantum eigensolver (VQE) method known as separable pair approximation (SPA) Kottmann and Aspuru-Guzik [2022a] and the ground state booster Wang *et al.* [2022b].The recent work Pathak *et al.* [2023] explored similar aspects of the aforementioned questions but for the near-optimal "non-heuristic" GSP method of Lin and Tong Lin and Tong [2020b]. Given the near-optimal provable performance guarantees of such methods, we choose to investigate heuristic ground state preparation techniques that stand to outperform these methods (though just empirically).

Here, we perform a detailed numerical analysis and resource estimation to compare heuristic ground state preparation techniques. Quantitatively, we aim to determine the degree to which different ground state preparation methods reduce the runtime of ground state energy estimation. This involves estimating the overlaps between the ansatz states and the ground state as well as estimating the number of quantum gates used to implement the circuits; as we will explain, both components play a role in determining the runtime reduction. As is common in fault-tolerant resource estimation Kim *et al.* [2022b]; Goings *et al.* [2022c]; Beverland *et al.* [2022]; Pathak *et al.* [2023], we will assume that the number of non-Clifford operations, specifically T-gates, governs the runtime of the quantum circuit and therefore measure runtimes in terms of circuit T-gate counts.

As a system of study, we use the 1-D hydrogen chains, which are good candidates for strongly correlated multi-electron systems Kottmann and Aspuru-Guzik [2022a]; Stair and Evangelista [2020]. Such systems capture many central themes of modern condensed-matter physics Motta *et al.* [2020] and the essential features of the many-electron problems in real materials Graner *et al.* [2019]. Interestingly, even for such simple models as linear hydrogen chains, research is still ongoing; only recently, fundamental ground state properties were computed in Ref. Motta *et al.* [2020].

This paper is organized as follows. In Sec. 5.2.2, we focus on the first question posed on how quantum GSP methods can improve small values of the HF overlap. In Sec. 5.2.3, we explore the performance of quantum GSP methods as a function of the system size. In Sec. 5.2.4, we analyze the speed-ups that quantum GSP methods could give for the task of GSEE. Finally, we present the conclusions and outlook in Sec. 5.2.5.

5.2.2 Quantum GSP performance over HF

In this section, we focus on the first question posed in the introduction: *To what degree can GSP algorithms improve the ground state overlap when the corresponding HF overlap has small values?* Both the SPA and booster algorithms have the potential to boost the overlap value even when the value of the initial HF overlap is very small, though the gate costs of the methods may need to increase as the overlap decreases Kottmann and Aspuru-Guzik [2022a]; Wang *et al.* [2022b].

The SPA algorithms are variations of the VQE algorithm optimized using a separable pair approximation (SPA), which assumes that a (closed shell) N-electron system could be described by a wavefunction of $N/2$ electron pairs. Each electron pair is represented by a wavefunction restricted to a disjoint subset of orbitals. Once combined with specific circuit compilation strategies, they can give classically tractable circuit classes with very short circuit depths. Specifically, the circuit depths are constant with the system size N and scale linearly with the basis size Kottmann and Aspuru-Guzik [2022a]. Moreover, as with other variational methods like adiabatic state preparation, the SPA does not necessarily require high overlap between its input state (e.g., Hartree-Fock) and the true ground state, as it can, in principle, prepare states orthogonal to Hartree-Fock due to its unitary nature.

The booster algorithm implements a function f (i.e., a Gaussian) of a Hamiltonian H, which suppresses the high-energy eigenstates of the Hamiltonian and enhances the low-energy ones in the expansion of an initial state. These functions f are referred to as boosters Wang *et al.* [2022b]. Even though some initial non-zero overlap is necessary for the booster algorithm to perform, it was demonstrated that it can convert an increase in circuit depth into an increase in overlap value, which in general, is not a feature of VQE-type algorithms. The aforementioned question becomes quite interesting for the booster algorithm if we limit the depth proxy D (or truncation level, see App. E.2 for more details) of the truncated Fourier expansion $f_D(H)$ of the booster function f to a certain value. Then, boosting the overlap becomes more challenging but also practical for implementations on early fault-tolerant quantum hardware Katabarwa *et al.* [2023].

In Fig. 5.4, we compare the behavior of the SPA algorithm to restricted HF for the system of H_6 with 8 spin-orbitals or qubits in an adapted basis (MRA-PNO basis) Kottmann *et al.* [2021b] for a range of bond distances (i.e. the spacing between hydrogen atoms). As already discussed, the SPA algorithm can improve the performance over the initial HF overlaps even when the HF fidelity has a very small value, i.e., 10^{-7}. Interestingly though, while the HF overlaps decrease exponentially as we increase the bond distance, the overlaps obtained with the SPA remain approximately constant. Even better, in the case of the booster algorithm, the squared overlaps are approximately equal to one for all bond distances.

For each bond distance shown in Fig. 5.4, we used different circuit variants of the SPA algorithm Kottmann and Aspuru-Guzik [2022a] and plotted the variants that give the maximum overlap. Specifically, for the H_6 molecule, we have the following variants SPA+GAS, SPA, SPA, SPA+GS, SPA+GS, and SPA+GS (referred to as "SPA+X" in Fig. 5.4) for the corresponding bond distances 1.5, 2.0, 2.5, 3.0, 3.5, 4.0 Å. The three variants supplement the SPA with unitary singles excitations: S adds singles from occupied to virtual orbitals, GS adds single excitations between all orbitals, and GAS approximates the singles by neglecting Pauli-Z operations in the Jordan-Wigner encoding.

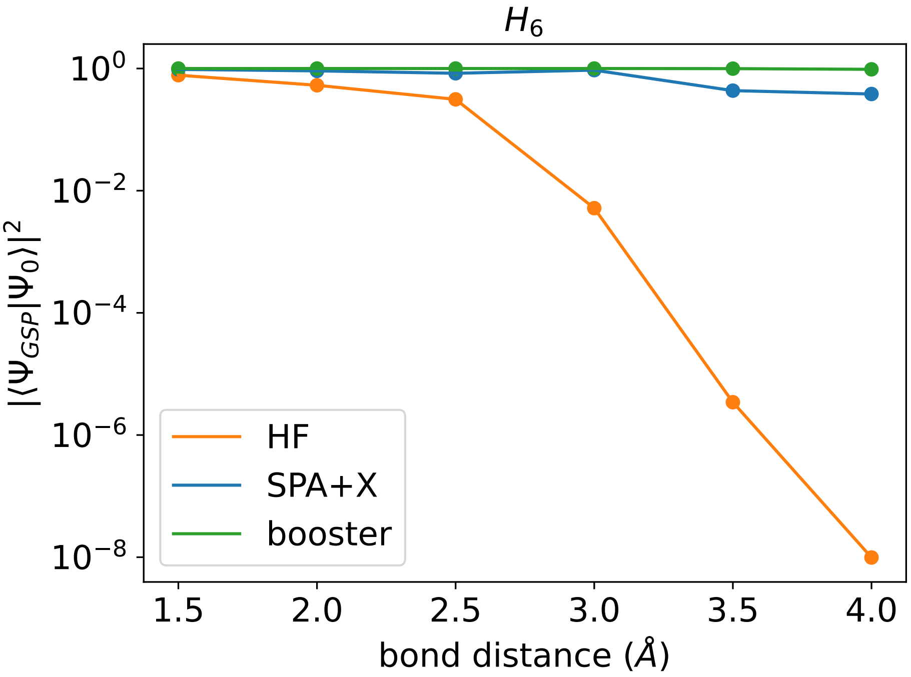


Figure 5.4 Here we plot the HF, SPA+X with approximates orbital rotations on different levels: $X \in$ GS, GAS, and booster $|\langle\Psi_{GSP}|\Psi_0\rangle|^2$ for the H_6 system with MRA-PNO basis set for an increasing bond distance.

5.2.3 Performance of quantum GSP algorithms versus system size

Next, we investigate the second question posed in the introduction: *Could quantum heuristic algorithms improve the ansatz overlap in terms of the exponential decrease with the system size?*

To this end, we explore the behavior of the hydrogen chains H_n with $R(H-H) = 3.0$ Å in terms of the overlaps as a function of the system size $n \in \{2, 4, 6, 8\}$. We use the MRA-PNO basis for H_n, which is closer to being complete compared to the minimal basis and is a better proxy for real-world use cases. Therefore, we have H_n for $n \in \{2, 4, 6, 8\}$ with $4, 8, 12, 16$ spin-orbitals or qubits respectively in an adapted basis Kottmann *et al.* [2021b].

In Fig. 5.5, we plot the HF, SPA-variants, and booster fidelities for the hydrogen chains H_n. In App. E.1, we plot the aforementioned fidelities for the same systems but with STO-3G basis set for comparison following Kottmann [2022] (see Fig. E.1). In Fig. 5.5, we have used the following SPA-variants (referred to as "SPA+X"): SPA, SPA+S, SPA+GAS, SPA+GS for n equal to $2, 4, 6, 8$, respectively.

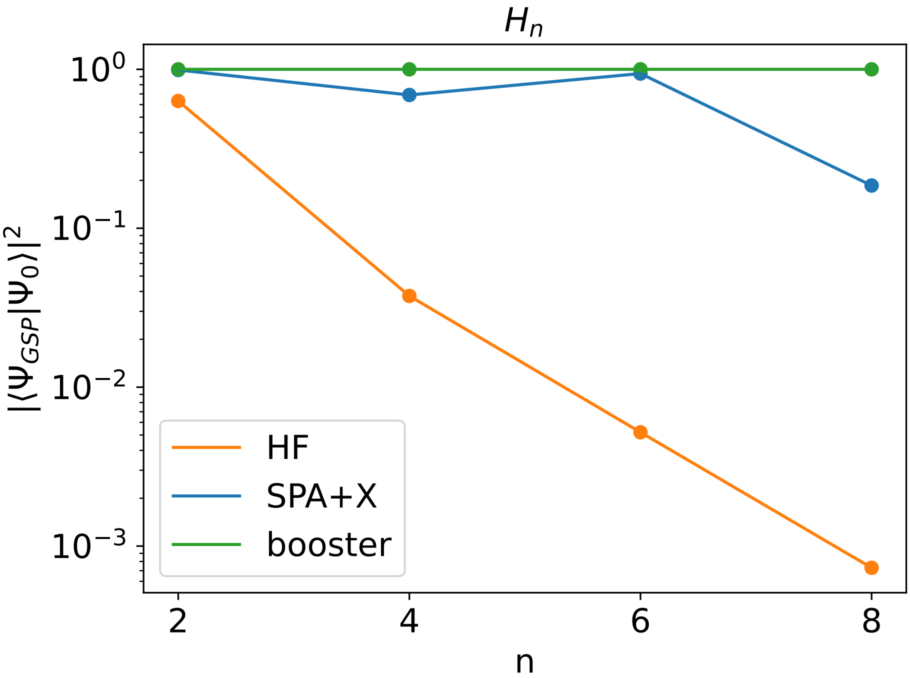


Figure 5.5 Here we plot the HF, SPA+X, and booster overlap squared for the H_n systems with MRA-PNO basis set and bond distance $R(H-H) = 3.0$ Å for increasing system size n.

While the HF fidelities drop exponentially with the system size n, the SPA algorithm gives more consistent performance. Interestingly though, the booster algorithm gives the maximum fidelity for all system sizes n. Potentially, for the case of $n = 8$, the SPA algorithm performance could be further improved.

Generally, the SPA algorithm optimizes the orbitals to give the best energy in the chosen electron pair model which was approximated through pair natural orbitals in the original work Kottmann and Aspuru-Guzik [2022a]. Some molecular instances though, require different shapes of orbitals to be optimal for SPA wavefunctions and the success of the orbital optimization depends on good initial guesses, which is, for example, relevant for $H_n = 8$ in Fig. 5.5.

Moreover, the SPA-variant could be further optimized since the SPA+GS depends on the initial guess of ansatz parameters, taken to be "random-close-to-zero" throughout this work, but with suitable intuition as in Kottmann [2022], the fidelity can be increased to around 90%. Extensive numerical analysis, to this end, goes beyond the scope of this work, which aims to analyze state-of-the-art procedures that can be applied in a semi-automatic way.

Another important factor for VQE algorithms relates to the number of individual measurements M that affect the runtime linearly and grow inversely proportional to the square of the total energy estimation error. This error can be expressed as $\eta + \epsilon$, where ϵ refers to the ansatz expressivity error, while η to the sampling error with $\eta \leq \epsilon$ Gonthier *et al.* [2022a]. In the worst case, $\eta = \epsilon$ which suggests that M grows as $1/(2\epsilon)^2$).

Generally getting the energy error within ϵ does not guarantee that the fidelity error will be within the desirable precision Mayer [2003]; Saad [2011]. But in certain cases, like the studied H_6 system we end up with an acceptable error in both the energy ($\epsilon = 17.9mHa$) and fidelity ($F = 0.94$) estimations. Specifically, the energy estimation of the VQE algorithm within $\epsilon = 17.9mHa$ is acceptable in scenarios where QPE will be applied and improve the energy estimation to within chemical accuracy. Next, the efficiency criteria introduced in Gratsea *et al.* [2022] are satisfied which renders the VQE algorithm studied here as an acceptable quantum GSP over HF. This is because the fidelity provides a high success rate and a significant speedup in the total runtime of the GSEE algorithm.

Usually, chemical accuracy $\epsilon_0 = 1.6mH$ is the target for the energy estimation error ϵ of a VQE algorithm, but as discussed in the example above a larger energy error, i.e. $\epsilon = 17.9mHa$, might be acceptable. In that case, we are afforded a significant reduction in the number of measurements $M_0/M = \epsilon^2/\epsilon_0^2 = 100$ required for a single energy evaluation of the VQE algorithm. The decreased sampling rates might negatively affect the gradient computations of VQE Sweke *et al.* [2020]; Gonthier *et al.* [2022a], but in the studied VQE algorithm the SPA angles could be optimized classically and avoid the aforementioned issue.

5.2.4 Speed-ups of quantum GSP algorithms over HF

In this section, we focus on the last question posed in the introduction: *Assuming the quantum GSP algorithms as subroutines of GSEE, what speed-ups could they offer?* To this end, we first apply the acceptability criteria introduced in the recent work of Gratsea *et al.* [2022]. According to this work, to determine whether to accept or reject a GSP over the HF we need to carefully examine the cost-benefit ratio, i.e., the resource cost needed to gain an increased overlap value.

In Table 5.1, we present the circuit depths of the studied SPA-variant circuits after compiling the unitaries containing single- and double-excitation gates to single-qubit Pauli rotations R and CNOT gates (see App. E.6). Then, each Pauli rotation is assumed to be implemented as a Clifford+T circuit following the suggestion in Ref. Ross and Selinger [2016] giving the total T-gate count of

$$\text{T-count} \approx 3 \times R \times \log_2\left(\frac{1}{\delta}\right), \tag{5.22}$$

where δ is the necessary precision per gate for operating R gates. As described in App. E.5, we choose δ through the failure tolerance δ_C of the GSP circuit. that we are willing to tolerate. If we set the circuit failure tolerance to $\delta_C = 0.001$ for all n, we get the respective values of δ (see Table 5.1 and App. E.5 for more details). Note that the main conclusions will not significantly depend on the value of the failure tolerance.

In Table 5.1, we also give the T-gate counts of the method among the SPA variants with the fewest T-gates per circuit. Moreover, we report the T-gate counts of the SPA+GS as a proxy for more advanced methods where explicit representations of the orbital rotations in the circuit are necessary, as for example in Kottmann [2022] or in case the following GSEE would require the Hamiltonian to be in a different orbital representation. As shown in table 5.1, the H_2 system remains on the SPA level as it is already represented exactly.

Next, we estimate the T-gate counts T_{GSEE} of the GSEE algorithm for the n system size by using the OpenFermion resource estimation module McClean *et al.* [2020]. Specifically, we get the estimated resources in terms of Toffoli gate count for single factorized QPE as described in Ref. Motta *et al.* [2021]; Berry *et al.* [2019]; Lee *et al.* [2021]. To convert the Toffoli gate count to T-gate we assumed Toffoli $= 4 \times$ T-gate Jones [2013].

Table 5.1 The lower (T_{SPA}) and upper (T_{SPA+GS}) bounds on the T-gate counts of the SPA circuits for H_n. Also, we give the T-gate counts of the GSEE algorithm (T_{GSEE}).

H_n	T_{SPA}	T_{SPA+GS}	T_{GSEE}
H_2	21	21	9.6×10^5
H_4	46	4.2×10^3	1.3×10^7
H_6	72	1.1×10^4	5.2×10^7
H_8	100	2.2×10^4	1.6×10^8

Comparing the T-gate counts of the SPA and GSEE algorithms shown in Table 5.1, we observe that $T_{GSEE} + T_{SPA+X} \approx T_{GSEE}$. Then, as discussed in Gratsea *et al.* [2022], the acceptability criterion becomes $1 < \left(\frac{\gamma}{\gamma_0}\right)^{\alpha+\beta}$, where γ and γ_0 correspond to SPA and HF overlap, respectively. Here γ^{β} is the scaling of the gate complexity of the ground state energy estimation circuit (excluding the state preparation) while γ^{α} is the scaling of the number of circuit repetitions needed to ensure the success for the given GSEE algorithm Dong *et al.* [2022a]. Given the criterion's dependence on the sum of these parameters and taking a few example algorithms from Dong *et al.* [2022a], we consider the GSEE algorithms to be represented by $\alpha + \beta = \{1, 2, 4\}$. For example, $\alpha + \beta = 1, 2$ and 4 correspond to the works of Lin and Tong [2020c]; Knill *et al.* [2007] and Lin and Tong [2022b], respectively. Generally, a lower sum indicates a more-performant GSEE algorithm. Figure 5.6 demonstrates that the SPA-variants are acceptable for all of the aforementioned values of $\alpha + \beta$ over the HF state preparation for the H_n systems shown in Fig. 5.5.

According to Gratsea *et al.* [2022] the speed-up ratio becomes $\frac{T_0}{T} = \left(\frac{\gamma}{\gamma_0}\right)^{\alpha+\beta}$, where T_0 and T refers to running the GSEE algorithm with HF and SPA-variants as subroutines for ground state preparation, respectively. In Fig. 5.6, we plot the aforementioned speed-up ratios for the values of $\alpha + \beta \in [1, 2, 4]$.

Next, we perform the same analysis, but with the low-depth booster as the GSP method. In that case, the circuit depth (measured as the number of accumulated controlled-exp $(2\pi i H)$ operations) is $2D$ where D is the depth proxy (or truncation level) of the truncated Fourier expansion of $f_D(H)$ (see App. E.2).

Given that the booster operation can be implemented with the linear combination of unitaries (LCU), we use a Trotter decomposition with one Trotter step to estimate the lowest cost in terms of Pauli rotations R and CNOT gates (see Table E.3). We can convert the Pauli rotations to T-gates using Eq. 5.22 (see $T_{K=1}$ in Table 5.2). The cost of the booster circuits in T-gate count is $T_B = 2DKT_{K=1}$ (see Table 5.2), where K is the number of total Trotter steps required. A more detailed analysis of the booster resource estimation can be found in App. E.2.

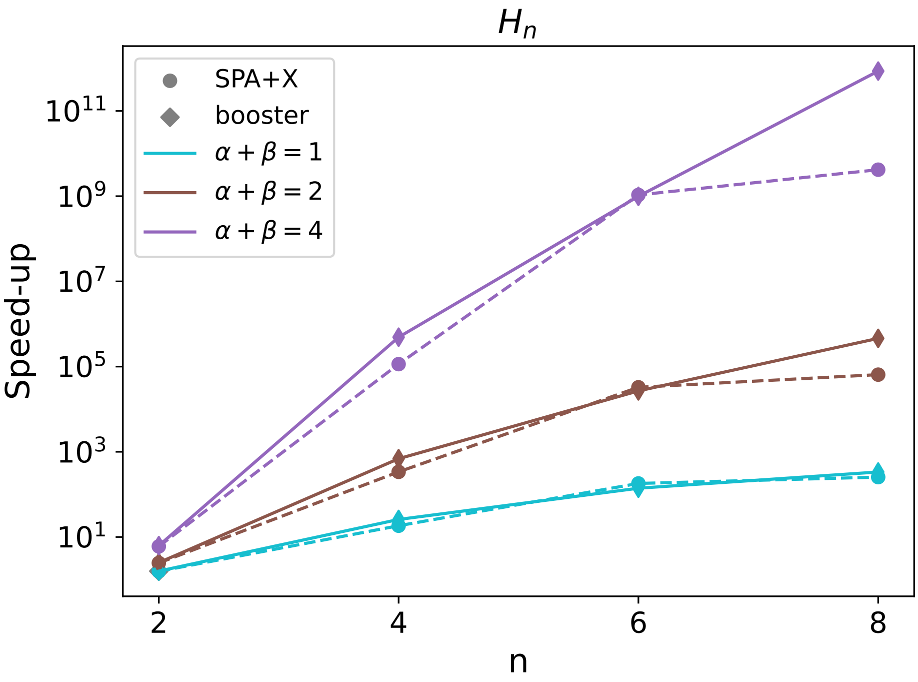


Figure 5.6 Here we plot the speed-up ratios T_0/T for SPA+X and booster algorithms over HF of H_n system for an increasing system size n and bond distance $R(H-H) = 3.0$ Å. The α and β parameters are scaling parameters for GSEE algorithms

The recent work of von Burg *et al.* [2021] empirically found that the required number of Trotter steps K for the 1-D Hydrogen chain is approximately equal to 10 and we discuss in more detail the applicability of this result in our case in App. E.3. In this work, we fix $D = 10$, since it is sufficient for the booster to give overlap values equal to 1.0 with the respective success probability P_{succ} shown in Table 5.2. The T_{GSEE} is the same as in Table 5.1.

Following the work of Gratsea *et al.* [2022], we can calculate the speed-up ratio $\frac{T_0}{T}$. Here we drop the $\tilde{O}$ by using the T-gate counts

Table 5.2 We present the important factors for the booster resource estimation analysis: the success probability (P_{succ}) and the total T-gate counts (T_B) along with the T-gate counts $T_{K=1}$ of the Trotter decomposition with one Trotter step $K = 1$ and of the GSEE algorithm (T_{GSEE}) for H_n.

H_n	P_{succ}	$T_{K=1}$	T_B	T_{GSEE}
H_2	0.46	9.3×10^2	1.9×10^5	9.6×10^5
H_4	0.027	5.6×10^4	1.1×10^7	1.3×10^7
H_6	1.1×10^{-4}	3.5×10^5	7.0×10^7	5.2×10^7
H_8	1.1×10^{-4}	1.3×10^6	2.6×10^8	1.6×10^8

presented in Table 5.2 for both the booster and GSEE algorithm. Thus, the acceptability criteria can be written as

$$\frac{\mathcal{T}_0}{\mathcal{T}} = \frac{\left(\frac{T_{GSEE}}{\tilde{\epsilon}\gamma_0^{\alpha+\beta}}\right)}{\left(\frac{1}{\gamma^{\alpha}}\left(\frac{T_B}{P_{succ}} + \frac{T_{GSEE}}{\tilde{\epsilon}\gamma^{\beta}}\right)\right)}, \tag{5.23}$$

where T_B and T_{GSEE} are the T-gate counts of the booster and GSEE algorithm with target accuracy $\tilde{\epsilon}$. In Fig. 5.6, we plot the speed-up ratios for the systems presented in Fig. 5.5 with the corresponding T-gate counts presented in Table 5.2 (see App. E.4 for the respective overlap values). According to the work of Gratsea *et al.* [2022], since all speed-up ratios are larger than one, the booster is acceptable over HF for all system sizes presented in Fig. 5.6.

5.2.5 Discussion and Outlook

We performed simulations on small strongly correlated systems H_n with an increasing bond distance $R(H-H)$ and system size n. These systems show a rapid decrease in fidelity between the restricted Hartree-Fock state and the ground state. On the contrary, both quantum GSP methods studied, i.e., SPA and booster, give fidelities close to 1.0. Similar performance is observed in Fig. 5.5, where we study different hydrogen chains with increasing system size $n \in [2, 4, 6, 8]$.

Following the work of Gratsea *et al.* [2022], we also focused on the cost-benefit ratio for the quantum GSP methods for the task of GSEE. In Fig. 5.6, we plot the speed-up ratios gained by using the SPA and booster algorithm over the HF for the studied H_n systems with increasing size n. The $\alpha + \beta$ determines the dependence of the GSEE algorithm on the ground state overlap γ, i.e. $(\gamma)^{-(\alpha+\beta)}$ Dong *et al.* [2022a]. Therefore, as suggested by Fig. 5.6, the smaller the value of $\alpha + \beta$, the smaller the speed-up ratio gained by the quantum GSP algorithm over the HF.

To put these results into context, recently there have been ongoing research on whether quantum GSP methods are necessary over classical ones. Different works have stressed the difficulties with classical GSP methods and the related problem of vanishing overlaps for certain chemical systems Kohn [1999]; Lee *et al.* [2022]. But other works have proposed that classical state preparation methods should not be a limiting factor in phase estimation even for large or strongly-correlated systems Goings *et al.* [2022c]; Tubman *et al.* [2018].

Our work suggests that quantum heuristic GSP methods could be beneficial over classical ones by reducing the runtime requirements of the GSEE algorithms in which they are used. Even for computationally affordable (i.e. requiring few resources and time to complete) strongly-correlated systems of intermediate size, i.e. H_n with $n \in [4, 6, 8]$, we observe significant speed-ups over HF method for both SPA and booster algorithm. Already for $n = 4$ we report an order of magnitude speed-up for both studied GSP methods, while for most instances of n and $\alpha + \beta$ we have many orders of magnitude runtime improvement.

Most literature that explores runtime improvements of quantum algorithms on quantum chemistry tasks has focused on large-scale systems and materials Kim *et al.* [2022b]; Lee *et al.* [2022]; Goings *et al.* [2022c]. However, recent works on classical chemistry methods establish the linear hydrogen chain as a benchmarking system for numerical simulations Motta *et al.* [2020]; Stair and Evangelista [2020]; Graner *et al.* [2019]. Thus, despite its simplicity, the hydrogen chain incorporates a rich set of physical and chemical properties Motta *et al.* [2020] and could be used as a benchmarking set to help develop quantum computing applications for quantum chemistry.

In principle, the success of the studied quantum GSP algorithms for the hydrogen chains could have been predicted but not the exact values of the scaling. For example, other systems that could exhibit similar behavior with the 1-D hydrogen chains are linear and cyclic organic molecules with pi-bonds (for example, alkenes or aromatic hydrocarbons C_nH_n).

There are several directions following this line of exploration that we leave for future work. First, it would be important to assess the GSEE runtime savings in cases where the HF state preparation may have an associated cost. In this paper, we assumed that HF state preparation is a cost-free operation on quantum hardware, which is valid in second quantization of the Hamiltonian but not necessarily valid with other approaches such as first quantization Su *et al.* [2021]. It would be important for future work to re-evaluate the conclusions of this paper using such methods where the Hartree-Fock preparation incurs a T-gate cost.

Second, the analysis could be extended to hydrogen chains with an increased system size N Fomichev *et al.* [2023] and with different bond distances and geometries Stair and Evangelista [2020]. The studied bond distance 3.0 for the linear hydrogen chain corresponds to a Mott insulator phase Motta *et al.* [2020]; Graner *et al.* [2019], and the correlations can be characterized by a spin-1/2 Heisenberg chain. Decreasing the bond distance below 1.7 Motta *et al.* [2020]; Graner *et al.* [2019] will allow us to study significant subjects in quantum material physics, such as metal-insulator transitions and magnetism.

As ground-state energy estimation is one of the most promising tasks for realizing on quantum computers, our work sheds light on how quantum heuristic GSP algorithms could reduce the runtime requirements of GSEE, and, therefore, make them more realistic for implementations on quantum hardware. Our work highlights that runtime improvements could be achieved for the task of GSEE by quantum heuristic GSP methods over HF for computationally affordable strongly correlated systems of intermediate size, like linear H_n with $n \in [4, 6, 8]$, which can capture interesting physical phenomena. This work aims to elucidate the landscape of methods that might someday be used to solve utility-scale problems in quantum chemistry.

5.3 Conclusions

This work is one of the first attempts to benchmark the performance of quantum and classical algorithms by incorporating both the resource cost (in terms of quantum circuit depth) and performance boost (in terms of overlap achieved between the prepared and true ground state of the studied system). This study performs a careful analysis of how quantum algorithms could perform in quantum chemistry computationally affordable strongly-correlated systems of intermediate size. Next, it would be interesting to perform the same analysis to larger quantum systems.

Chapter 6

Early fault-tolerant quantum computing

The previous chapter focused on the performing a resource estimation on two different quantum algorithms for the task of ground state energy estimation (GSEE). Generally, extensive research is performed to evaluate the capabilities of quantum computing for specific applications and the required quantum computation power that will be necessary towards this end. This computation power of the quantum hardware is usually labeled as Noisy-Intermediate quantum computing (NISQ) or fault-tolerant quantum computing (FTQC) devices. NISQ refers to the quantum computing devices available today with approximately 100 qubits and 100 circuit depth, while FTQC corresponds to quantum computing devices with full quantum error correction capabilities. But what seems necessary for further advancing the field of quantum computation in a timely manner is bridging this dichotomy between NISQ and FTQC. To this end, the regime of early fault-tolerant quantum computing comes into play.

This work aims to discuss how research could drive the transition from NISQ to FTQC:

NISQ : In the seminal paper from J. Perskill in 2016, the term noisy intermediate scale quantum computers (NISQ) was introduced. Even though the definition of the term was not very strict it characterized the quantum devices with a small number of qubits and circuit depth where quantum error correction is too costly to be implemented on hardware. Since then, the NISQ era has seen significant progress in reducing the error rates characterizing the quantum hardware. There are numerous experimental demonstrations that current quantum devices could still perform certain tasks in the regime of small number of qubits and shallow circuit depth despite all the problems that arise. Of course, the pressing question is whether this era could give practical quantum advantage, i.e. showcase that these devices are capable of performing tasks with, for example, industrial relevance by either aiding classical quantum computers or targeting tasks beyond the capabilities of classical computation. Interestingly, this now brings the question back to quantum algorithm developers to come up with interesting problems that current quantum devices could tackle in the range of 100 qubits and 100 circuit depth.

FTQC : The seminal papers of P.Shor and others, introducing the Shor's algorithm and quantum error correction, revolutionized the understanding of the capabilities of quantum computation. Unfortunately, the cost of these operations on quantum hardware are beyond the capabilities of NISQ devices and further research will be necessary in the span of the following decades to make this a reality. Interesting results have been discussed on the potential capabilities of FTQC from complexity theorem and resource estimation analysis to further quantify the performance of these devices and the potential quantum advantages that could be unlocked in this regime.

NISQ-to-EFTQC-to-FTQC : But what seems necessary for further advancing the field of quantum computation in a timely manner is bridging this dichotomy between NISQ and FTQC. To this end, the regime of early fault-tolerant quantum (EFTQC) computing comes into play. The EFTQC regime could help better understand how the transition from NISQ to FTQC could be realized in practice. For example, studying the interplay between QEM and QEC can characterize the transition from NISQ-to-EFTQC. Introducing realistic noise models that capture the performance of quantum devices can guide algorithm developers in exploiting the capabilities of quantum hardware. Another interesting direction to further work on is the transition from EFTQC-to-FTQC. To this end, benchmarking quantum algorithms that traditionally belong to EFTQC and FTQC could help understand what problem sizes in terms of number of logical qubits each algorithm could tackle for the same number of physical qubits and error correction scheme. The EFTQC era could help bridge the gap between NISQ and FTQC regimes and the corresponding efforts in these two fields of quantum computation.

The original results presented in this chapter are based on Katabarwa *et al.* [2023], a joint collaboration with A. Katabarwa, A. Caesura and P. D. Johnson.

6.1 Introduction

Quantum computers were first proposed to efficiently simulate quantum systems Feynman [1982b]. It then it took about a decade before it was discovered that quantum phenomena, such as superposition and entanglement, could be leveraged to provide an exponential advantage in performing tasks unrelated to quantum mechanics Deutsch and Jozsa [1992]. Although of no practical use, the Deutsch–Jozsa algorithm sparked interest in using a quantum computer to perform other tasks beyond simulating quantum systems Grover [1997]; Bernstein and Vazirani [1993], the most famous case being Shor's algorithm **?**. Around the same time the ground-breaking discovery of quantum error correcting codes (QECC) Shor [1995]; Steane [1997]; Laflamme *et al.* [1996]; Knill and Laflamme [1996]; Calderbank *et al.* [1997] set the stage for practical quantum computing. This showed that errors due to faulty hardware could not only be identified but also corrected. Two pieces of the puzzle were left, namely:

1. Could quantum computation be done in a fault tolerant manner i.e. could error-corrected qubits perform better than physical qubits?

2. Can one rigorously prove the existence of a threshold[1] below which error can be reduced exponentially in the time and memory overhead cost?

The first piece of the puzzle was tackled by Peter Shor Shor and later, building on his work, threshold theorems were proved assuming various kinds of error models Aharonov *et al.* [2006]; Aharonov and Ben-Or [2008a]; Kitaev [2003]. For a specific quantum error correcting code and a noise model it is then left to prove and find error thresholds, with early works being Aliferis and Leung [2006]; Aliferis and Preskill [2008]; Aliferis *et al.* [2008, 2006]; this continues to be an active area of research Fowler *et al.* [2012]; Kovalev and Pryadko [2013]; Breuckmann and Eberhardt [2021]; Cohen *et al.* [2022].

Meanwhile on the hardware side, astonishing progress has been made across various modalities (e.g. superconducting, ion trap, photonic, etc.) in terms of extending qubit coherence times and improving entangling operations Wintersperger *et al.* [2023]; Shi [2022]; Debnath *et al.* [2016]; Hanneke *et al.* [2009]; Monroe *et al.* [1995]; Bao *et al.* [2018]; Chen *et al.* [2014]. Driven by such advances, a watershed moment occurred in 2016 when IBM put the first quantum computer on the cloud giving the public access to quantum computers. This event spurred widespread interest in finding near-term quantum algorithms that did not need the full machinery of fault tolerance. These algorithms first formulate the problem as a solution to the ground state of some Hamiltonian store a trial ansatz on the quantum processing unit (QPU) and use a classical optimizer to find

[1] A further requirement is that realistic assumptions be made on the noise model.

the optimal parameters. The variational principle guarantees that the optimized parameters will produce a state whose energy upper bounds that of the target Hamiltonian. These so called hybrid *quantum/classical* algorithms allow one to use short depth quantum circuits and reduce the need for high quality quantum coherence. They have found application in areas of quantum chemistry Peruzzo *et al.* [2014b], machine learning Liu and Wang [2018b]; Dallaire-Demers and Killoran [2018] and optimization Farhi *et al.* [2014]. Another watershed moment occurred when the Google Quantum AI team along with collaborators announced their achievement of so-called "quantum supremacy" Arute *et al.* [2019a]; they argued that their hardware accomplished a computational sampling task far faster than possible with available supercomputers.

Despite this progress, there is still need to reduce errors and the area of quantum error mitigation arose as attempts were made to meet the needs of these applications Temme *et al.* [2017]; Czarnik *et al.* [2021]; Giurgica-Tiron *et al.* [2020]; Huggins *et al.* [2021]; van den Berg *et al.* [2023]. This way of using a QPU is what is characteristic of the so called *NISQ era* Preskill [2018]. Although there is no strict definition of what constitutes a NISQ device it can generally be assumed that NISQ devices are too large to be simulated classically, but also too small to implement quantum error correction. IBM's work Kim *et al.* [2023b] in some sense is the true dawn of the NISQ era i.e a quantum device where error mitigation is important and classical simulation is hard. The flurry of work Kechedzhi *et al.* [2023]; Tindall *et al.* [2023]; Begušić and Chan [2023] immediately arose pushing classical methods of simulation and claiming to reproduce IBM's results. This is a new phase in which NISQ devices will be put to the test by state of the art classical simulators and vice-versa. This back and forth will not last long as Hilbert space for quantum systems grows exponentially and the NISQ device will be the only viable simulation approach.

But an important question remains and in a very obvious sense the elephant in the room is, "are NISQ devices and NISQ algorithms up for the task of realizing quantum advantage at utility scale?” Work has been done in quantum chemistry where the problem can be precisely asked, for example, in finding the ground state of large molecules. The best estimates so far for resource estimates suggest the variational quantum eigensolver (VQE) is not up to the task Gonthier *et al.* [2022b]. Other work suggests a possible quantum advantage for the quantum adiabatic optimization ansatz (QAOA) Farhi and Harrow [2016]; Shaydulin *et al.* [2023]; Lykov *et al.* [2023] in optimization but it remains to be seen whether these claims can be confirmed in the presence of noise at scale.

Given these roadblocks, should our attitude be to wait for fully fault tolerant devices? An area of research offers an intriguing possibility; we are offered a trade-off, we require fault tolerant quantum computing but the ability to run smaller quantum circuits at the cost of requiring more sampling for the quantum device. Such a trade-off has been the focus of a substantial amount of research in the past few years Wang *et al.* [2019]; Parrish and McMahon [2019]; Wang *et al.* [2021]; Giurgica-Tiron *et al.* [2022b]; Klymko *et al.* [2022]; Wang *et al.* [2023a]; Dong *et al.* [2022b]; Wang *et al.* [2022c]; Ding and Lin [2022b]; Kirby *et al.* [2023] However, in a regime where we are able to arbitrarily scale the number of physical qubits while maintaining quality fault-tolerant protocols, such a trade-off would not be favorable; by increasing the circuit size using methods such as quantum amplitude amplification Cleve *et al.* [1998], the additional overhead of efficient fault-tolerant protocols is negligible compared to the overall reduction in runtime. Accordingly, such a trade-off would be better suited to a setting in which the efficiency of fault-tolerant protocols worsens with increasing system size. If the ability to scale the number of physical qubits (i.e. the “scalability”) is compromised by a worsening of the operations, then this diminishing returns will, in turn, limit size of problems that can be solved. Such a regime of computation has been referred to as *early fault tolerant quantum computing* (EFTQC) Campbell [2021], a natural successor to the NISQ era. A field of research has emerged recently where the proposed quantum algorithms enable this "circuit-size vs sample-cost" trade-off Wang *et al.* [2019]; Zhang *et al.* [2021]; Wang *et al.* [2021]; Lin and Tong [2022a]; Wan *et al.* [2022]; Giurgica-Tiron *et al.* [2022b]; Dong *et al.* [2022b]; Ding and Lin [2023]; Wang *et al.* [2022a, 2023b]. Two questions are then placed before us:

1. Will this regime of limited-scale quantum computers exist in a meaningful way?

2. If so, will we be able to unlock intrinsic quantum value at scale in this regime?

The ultimate answers to these questions will depend on hard-to-predict factors including hardware, QECC and quantum algorithm advances, and improvements in competing classical hardware and algorithms. Rather than predicting the timeline of these advances, we propose a quantitative framework to track their progress. Figure 6.1 depicts the landscape in which this framework assesses the ability of a given hardware vendor to supply useful physical qubits, transitioning from NISQ to EFTQC to FTQC.

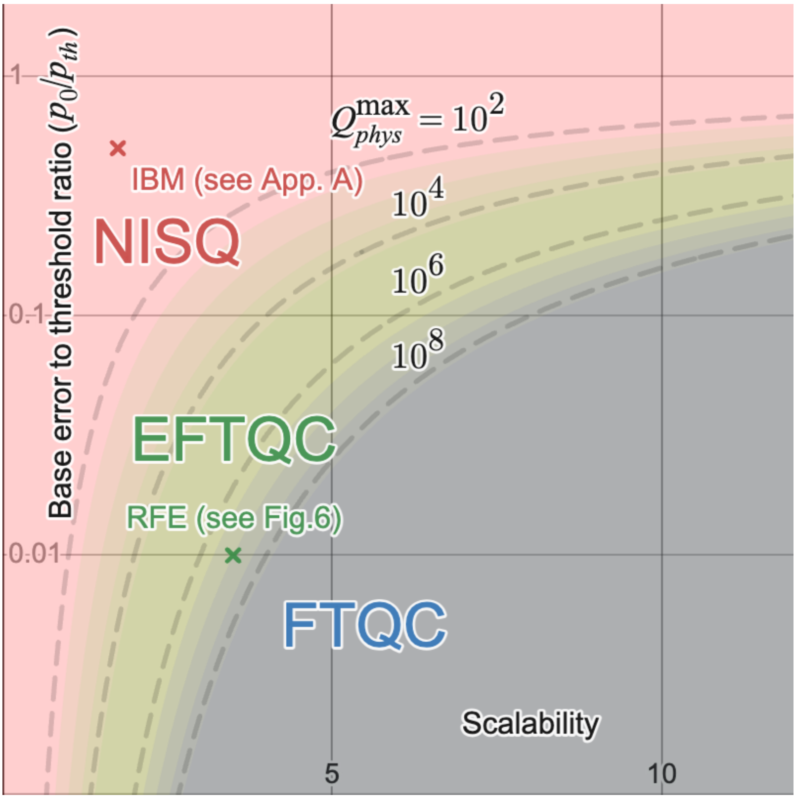


Figure 6.1 This figure roughly demarcates the regimes of NISQ, EFTQC, and FTQC according to the scalability model introduced in Section 6.2. The vertical axis quantifies the base error rate (i.e. that achievable for a single-qubit), while the horizontal axis quantifies the ability of the architecture to maintain low error rates as it is scaled (i.e. its *scalability*). Contours indicate the maximum physical qubit number that the architecture is warranted in scaling to as predicted by the scalability model of Equation 6.3. The NISQ-to-EFTQC transition is characterized by having enough qubits to implement fault-tolerant non-Clifford operations (e.g. T factories), while the EFTQC-to-FTQC transition is characterized by the ability to accommodate very large problem instances (e.g. encoding 10,000 logical qubits using in 10^9 physical qubits). The red x corresponds to data presented in Section F.1, which estimates that a hardware vendor of today (IBM) has a scalability of 1.75 with $p_0 = 0.005$. An editable version of the plot can be accessed here: https://www.desmos.com/calculator/9iphmmdjfp

To address the first question we propose a very simple model (see Equation 6.3) to quantitatively discuss these regimes. This simple model describes how the quality of elementary quantum operations degrade as system size is increased; that is, we model the physical gate error rate as a function of physical qubit number. We dub this model the *scalability* of a device. For the second question, we quantify how recently-developed algorithms can extend the "reach" of quantum computers with limited *scalability*. This is an important step towards understanding what value such methods can provide. Two results that we will establish (see Equations 6.10 and 6.12) are that, according to the scalability model, the optimal number of physical qubits to use in the architecture is the following function of scalability parameter s

$$Q_{\text{phys}}^{\text{opt}} = \frac{1}{e^2}\left(\frac{p_{\text{th}}}{p_0}\right)^s \tag{6.1}$$

and the maximum problem size, measured in terms of the largest number of logical qubits, is predicted to be

$$Q_L^{\max} \approx \frac{\left(\frac{p_{\text{th}}}{p_{\text{phys}}}\right)^s}{2e^2s^2\beta^2\ln\left(\left(\frac{A\alpha}{p_C}\right)^{\frac{1}{\beta}}\frac{\left(\frac{p_{\text{th}}}{p_{\text{phys}}}\right)^s}{8e^2s^2\beta^2}\right)^2}. \tag{6.2}$$

The various parameters are defined in Section 6.2.2. We will ultimately explore so-called "EFTQC algorithms", which enable an increase in $Q_L^{\max}$. We will explain how these expressions show 1) the importance of the scalability parameter s in governing the capabilities of a quantum hardware vendor and 2) the role played by the "fault-tolerance burden factor" $\frac{A\alpha}{p_C}$, that combines gate count, error correction, and algorithm robustness parameters. These elucidate multiple ways to improve a quantum computation towards solving utility-scale problem instances in the finite-scalability regime.

6.2 Modeling Early Fault-tolerant Quantum Computations

6.2.1 Introduction to the scalability model

In this section we establish and discuss the precise sense in which a device can be an *early fault tolerant device*. We first note the tension in the very phrase *early fault tolerance*. Fault-tolerance evokes the ability to ensure efficient suppression of error despite the use of faulty operations Shor. The string of results Aharonov and Ben-Or [2008b]; Knill and Laflamme [1996]; Aliferis *et al.* [2008]; Aharonov *et al.* [2006] collectively known as the threshold theorems show that *in principle* this can be achieved. In fact thanks to these results we know Aharonov *et al.* [2006] that under quite general assumptions such as allowing for long range correlations of noise and non-Markovianity, fault-tolerance is still possible. These foundational works would put the threshold error rate around 10^{-5} to 10^{-6}. However, more optimistic threshold predictions have been made using numerical investigations Zalka [1996]. For the surface code Dennis *et al.* [2002], which is a leading contender for practical quantum computing AI [2022], such simulations have led to the prediction of quite optimistic thresholds of $\sim 1\%$ Wang *et al.* [2011], which have also been argued for analytically Fowler [2012]. On the other hand, numerical thresholds are based on particular assumptions of noise and error that cannot fully capture the complexity of quantum architectures at scale. For example, an important assumption is that a single number can be used to capture the performance of operations and that this single number remains constant as larger code distances[2] are used Wang *et al.* [2011]. Such thresholds have become the established targets for hardware developers Xue *et al.* [2022]; Blume-Kohout *et al.* [2017];

[2] In the surface code, the degree of resilience to error is controlled by the size of the two-dimensional grid of qubits used to encode each qubit. The minimal number of single-qubit errors needed to cause a logical error, or the code distance, is the diameter of the two-dimensional grid of qubits.

Postler *et al.* [2022]; Egan *et al.* [2020].

The "early" in "early fault tolerance" on the other hand suggests some kind of limited ability to achieve fault tolerance i.e using polynomial amount of resources to achieve exponential error suppressionGottesman [1997]; Raussendorf *et al.* [2006]. This tension is what lies behind the motivation for this work.

A key insight towards resolving this tension is to realize what we might call the *scalability requirement*:

> *In order to reap the benefits of being below any threshold, an approach to building a quantum architecture must be able to maintain each operation below the threshold error rate as larger and larger architectures are built.*

The failure to achieve the scalability requirement implies the existence of scale dependent errors. To motivate where these scale dependent errors might come from we consider the general setup used to prove the threshold theorem. It is assumed we have the following Hamiltonian as

$$\mathcal{H} = \mathcal{H}_S + \mathcal{H}_B + \mathcal{H}_{SB},$$

where $\mathcal{H}_s$ is the hamiltonian governing the evolution of the system which for our discussion can be the evolution corresponding to implementing the quantum gate, $\mathcal{H}_B$ governs the evolution of some bath and $\mathcal{H}_{SB}$ entangles the bath with the qubits in the computation. The scale dependent errors arise from the engineering details involved in implementing $\mathcal{H}_S$ as larger and larger chips are developed. These engineering problems can't be completely inserted into $\mathcal{H}_{SB}$ and yet would ultimately impact how easily we could stay below threshold as we try to scale up. For fixed frequency qubits in superconducting architectures the issue of "frequency crowding" affects the quality that any single two-qubit gate can achieve Hutchings *et al.* [2017]. The number of frequencies that must be avoided when implementing the cross resonant gate, increases as the number of qubits increases in the chip; this makes targeting the required frequency harder and harder as you scale up. Another scale dependent engineering difficulty can arise from unwanted interactions between control lines going into the chip. The calibration of these pulses is partly a classical problem that gets more complicated and cumbersome as the chip gets larger. The issue of "cross mode coupling" at ion-traps will affect the fidelity of the gate Leung and Brown [2018]; Liang *et al.* [2023a], where the target has a specific motional mode but unwanted couplings destroy the quality of the gate. The problem has a classical component that scales with the number of qubits. In the above cases the physics of accurate addressability of a qubit or pairs of qubits is a problem that becomes harder with increasing number of qubits, and thus, affects the quality of the gate operation. Recent works have explored the consequences of scale dependent errors which would most likely arise from the limited resources to control qubits and design good quality operations Fellous-Asiani *et al.* [2021]; Fellous-Asiani [2021].

It is reasonable to believe that the assumption of scale-independent error rates may eventually become effectively true on account of modularity, as future quantum architectures will likely be made from repeated modular components. And while the holy-grail of (effectively) scale-independent, sub-threshold error rates may someday be realized, quantum architectures will necessarily undergo a transition from today's scale-dependent error to the future of scale-independent error. We will take this transition to be the defining characteristic of early fault-tolerant quantum computing.

Ultimately, this investigation is motivated by wanting to understand the prospects of using early fault-tolerant quantum computers to solve utility-scale problems. We take such machines to be characterized by a non-negligible degree of scale-dependent error. The standard approach to predicting the performance of fault-tolerant architectures for utility-scale problems is to assume that the error is scale independent Kim *et al.* [2022a]; Goings *et al.* [2022c]; Beverland *et al.* [2022]. Therefore, our approach will be seen as 1) a generalization that incorporates both the scale-independent and dependent settings and 2) an attempt to bridge the observed scale-dependence of error in today's devices with the hoped-for scale-independence of error in future quantum architectures. We expect that the degree of scale-dependence will inform the capabilities of the architecture being modeled. Furthermore, scale-dependent error may warrant the development and use of quantum algorithms that are suited to this limitation. These considerations motivate the main question pursued in the remainder of this manuscript: *how does the degree of scale-dependent error determine the capabilities of early fault-tolerant quantum computers?* Next, we introduce a model to capture the degree of scale-dependent error.

We start by describing the particular setting in which we model scale-dependent error. Our model will center around the concept of *scalability*, the ability to maintain low error rates (e.g. sub-threshold) as larger architectures are requested. Our setting and model are driven by the need to answer the question: for a *series* of quantum computations of increasing size, how well will a hardware vendor be able to service the request to run the quantum computations. Accordingly, we will not consider the capabilities of a single quantum device or a single quantum architecture, as the hardware vendor might have several architectures to service computations of various sizes. Furthermore, we will not consider the capabilities of the hardware vendor as they improve over time, as our hypothetical test is used to assess capability at one moment in time.

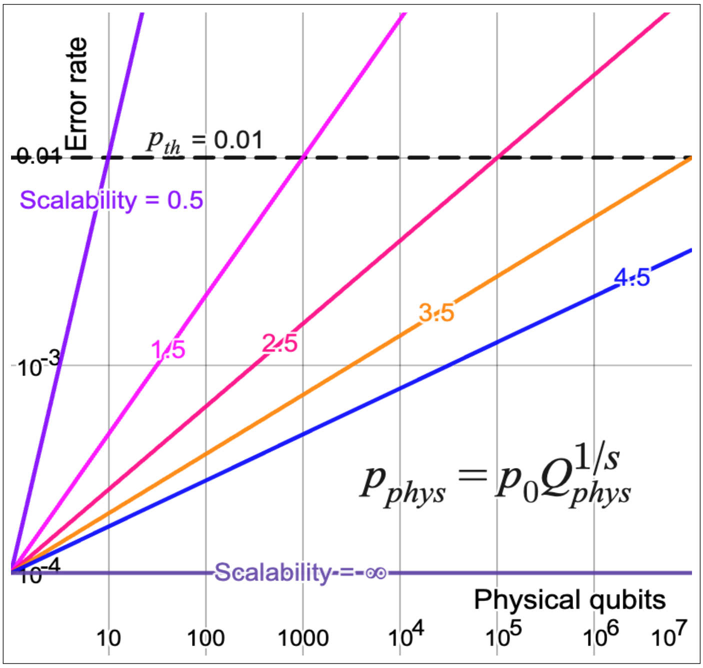


Figure 6.2 The concept of scalability captures the ability of a quantum architecture to maintain low physical error rates as the number of physical qubits of the architecture is increased. This figure shows the scalability profiles of different quantum architectures given by the scalability model (Eq. 6.3) for different scalability values ($s = 0.5, 1.5, 2.5, 3.5, 4.5, \infty$) and base error rate $p_0 = 10^{-4}$. A finite scalability implies that beyond a certain physical qubit size, the architecture cannot maintain physical error rates below the error threshold (p_{th}) of the fault-tolerant protocol. An editable version of the plot can be accessed here: https://www.desmos.com/calculator/jlmbygcqrp

In order to make this quantitative, we can consider a *scalability profile*: an empirically derived function that reports the worst-case error rate among the elementary operations of the device as a function of the requested number of physical qubits. For the case of today's IBM devices, we present data on their scalability profile in Appendix F.1. In lieu of scalability profile data for future quantum vendors, we propose a simple parameterized model for this function

$$p_{\text{phys}}(Q_{\text{phys}};\mathcal{V}) = p_0 Q_{\text{phys}}^{1/s}, \tag{6.3}$$

where Q_{phys} is the number of physical qubits in the architecture and $\mathcal{V}$ labels the particular hardware vendor that is providing the qubits at any time. Parameters p_0 and s capture the base error rate and the "scalability", respectively. It is helpful to view this model as a power-law fit of a scalability profile. In Appendix F.2 we investigate the more optimistic case of a logarithmic model. The case of $s = \infty$ corresponds to scale-independent error, or infinite scalability, while any finite value of s corresponds to the case of finite or limited scalability.

As we will show in the next section, in the context of fault-tolerant quantum computing, a finite scalability will result in a finite limit on the number of physical qubits being used before fault-tolerant protocols yield diminishing returns. We then explain how this limit on physical qubit number places a limit on the problem sizes that the architecture can accommodate. Importantly, all of these considerations apply in the setting where fault-tolerant protocols are being used. This differs from the setting assumed for NISQ quantum computing Preskill [2018], where physical qubits instead of logical qubits are used for computation. Before moving to the next section, we provide some perspective on the transition from the NISQ regime to the EFTQC regime. Specifically, in the rest of this subsection we estimate the minimal number for Q_{phys} in an EFTQC computation assuming a simple surface code architecture.

The total number of physical qubits for a computation can be written as $Q_{\text{phys}} = Q_{\text{comp}} + Q_{\text{MSD}}$ where Q_{comp} is the number of physical qubits used to compute (i.e. storing and routing the logical data) and Q_{MSD} are the physical qubits used for magic state distillation. To calculate the minimum number of qubits required for QEC, we will set $Q_{\text{comp}} = 2(d+1)^2$ Goings *et al.* [2022c] corresponding to single surface code logical qubit and pick the smallest distillation widgets which give an improvement on the error rate.

The most efficient distillation widgets known in the surface code

Quality of operations	Factory name	p_{phys}	Q_{phys}	p_{out}	$Q_{\text{min, EFTQC}}$	p_L
High	$(15\text{-to-}1)_{5,3,3}$	10^{-4}	522	4.7×10^{-6}	554	10^{-5}
Lower	$(15\text{-to-}1)_{7,3,3}$	10^{-3}	810	5.4×10^{-4}	842	10^{-3}

Figure 6.3 A table containing the two smallest possible magic state distillation factories given by Litinski [2019]. p_{phys} is the physical error rate, Q_{phys} is the number of physical qubits required to create the factory, p_{out} is the probability that the output state magic state is incorrect, $Q_{\text{min, EFTQC}}$ is a rough lower bound on the number of qubits in an EFTQC calculation, and p_L is the logical failure rate in that lower bound calculation. Note that in the case of superconducting qubits, the lower bound EFTQC example does not decrease the logical error rate p_L.

are given in Litinski [2019]. We have listed the smallest of these in Figure 6.3 (note that these do not give much of an improvement over the physical error rate). An important property of the magic state injection process is that it cannot have error rates which are less than the current logical level. Thus a magic state which is injected into a code with error rate p_L can have at most error rate p_L. If we have a single logical qubit in the smallest non-trivial surface code (i.e. distance 3), then this minimum viable example of EFTQC will be at least 540 and 826 qubits for lower and high-quality operations, respectively.

Note that the magic state distillation factory dominated the number of qubits. As a result, the FTQC community has put a lot of work into decreasing the size of factories Litinski [2019], improving injection protocolsGavriel *et al.* [2022], or eliminating distillation entirely Akahoshi *et al.* [2023]. One would expect that the first EFTQC demonstrations will employ many of these techniques rather than the "pure FTQC" calculation presented above. In a more careful calculation to estimate a lower bound for the EFTQC range, one may want to take such techniques into account and calculate Q_{phys}. Refining this estimate to clarify and lower the NISQ-to-EFTQC transition is important future work.

6.2.2 Example: quantum phase estimation compiled to the surface code

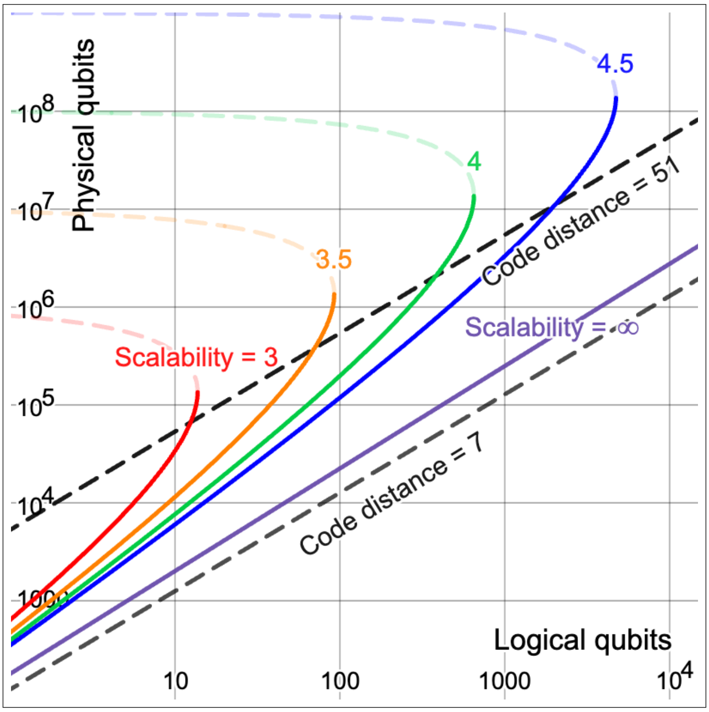


Figure 6.4 The scalability model of Eq. 6.3 predicts that, for each finite value of scalability parameter s, there is a maximum problem instance size that can be accommodated by the architecture. Each curve is a contour in the Q_{phys}-Q_L plane of a solution to Eq. 6.9 for a particular value of the scalability parameter s (3, 3.5, 4, 4.5, ∞). The remaining parameters of Eq. 6.9 are set to $p_{th} = 10^{-2}$, $p_0 = 10^{-4}$, $\alpha = 4.12 \cdot 10^9$, $\beta = 0.515$ following Table II in Campbell [2021]. The transition from solid to faded dashed curves occurs when the physical qubit number reaches $Q^{\text{opt}}_{\text{phys}} = Q^{\text{max}}_{\text{phys}}/e^2$, beyond which increasing the code distance leads to diminishing returns. The diagonal black dotted lines show the physical qubit count for two fixed code distances: 7 (small distance) and 51 (large distance). Note that code distance is discrete, which, if taken into account, would result in the contours jumping from one fixed-code-distance line to the next. However, we have chosen to allow for the distance parameter to be continuous, for ease of viewing the trends of the contours. An editable version of the plot can be accessed here: https://www.desmos.com/calculator/7mbziuf8gd

In the preceding subsection, we introduced Eq. 6.3 as a model for how physical operation error rates might increase with system size. To understand the implications of this model, we work through the example of using the quantum phase estimation (QPE) algorithm Cleve *et al.* [1998] to solve the phase estimation task. The task of phase estimation is to estimate the eigenphase of a unitary operator U with respect to an eigenstate $|\psi\rangle$ assuming access to circuits that implement c-U and prepare $|\psi\rangle$. We review how to estimate the quantum resources required to perform this task under the scalability model and compare these to the ideal model case (i.e. $s \rightarrow \infty$).

A fault-tolerant resource estimation answers the question: how many physical qubits are needed per logical qubit to ensure that the logical error rates are sufficiently low to make the algorithm succeed (with some probability)? To answer this, we must 1) determine what logical error rates the algorithm deems as "sufficiently low" and 2) establish the relationship between logical error rate and quantum resources.

For 1), the QPE algorithm will succeed with sufficiently high probability as long as the total circuit error rate is below some value p_C. We will set $p_C = 0.1$, noting that, in the literature, this tolerable circuit error rate varies from 0.1 Goings *et al.* [2022c] to 0.01 Kim *et al.* [2022a], but can be made lower using alternative algorithms Kimmel *et al.* [2015]; Kshirsagar *et al.* [2022]; Li *et al.* [2023]. This tolerable circuit error rate, along with the number of operations per circuit, lets us bound the tolerable operation error rate. The quantum circuit will ultimately be compiled into a set of logical operations that are implemented using fault-tolerant protocols (e.g. initialization of $|0\rangle$, measurement in the computational basis, H gate, CNOT gate, and T gate). We define G_C to be the number of elementary logical operations (including idling[3]) used by the circuit. To ensure that the

[3] Especially in the case where the quantum computation is rate-limited by magic state distillation, the computational qubits would be required to idle without accruing error while waiting for T gates or Toffoli gates to be teleported into the computation.

circuit error rate is less than p_C, it suffices[4] to ensure a logical error rate of $p_L \leq p_C/G_C$ (by the union bound).

For the quantum phase estimation algorithm, G_C is determined by the target accuracy and the number of operations per c-U. To yield an estimate of the phase angle to within ϵ of the true value requires using a circuit with $1/\epsilon$ applications of c-U Brassard *et al.* [2002]. For our purposes we assume a model for G_C by fitting data in Table II of Campbell [2021] to the following power law, where the ϵ is set to be approximately half a percent of the total system energy,

$$G_C = \alpha Q_L^{\beta}, \tag{6.4}$$

yielding $\alpha = 4.12 \cdot 10^9$ and $\beta \approx 0.515$. Thus, the algorithm success is ensured (with high probability) by

$$p_L \leq \frac{p_C}{\alpha Q_L^{\beta}}. \tag{6.5}$$

For simplicity, we'll assume that the number of logical qubits needed for magic state factories are accounted for in this model (see notes in the desmos plot of Figure 6.4 for details of the assumptions and the relevant references) and we will assume that the physical qubit overhead is captured by the code distance used for the data qubits (though the factories typically have multiple layers of concatenation with differing code distance).

For 2) we will assume a model of error suppression based on simulations of the surface code in Fowler and Gidney [2018]. This model is

$$p_L = A(p_{\text{phys}}/p_{\text{th}})^{(d+1)/2}, \tag{6.6}$$

where Fowler and Gidney [2018]; Goings *et al.* [2022c] estimates $A = 0.1$ and $p_{\text{th}} = 0.01$. The number of physical qubits used to encode

[4] In the case where the tolerable circuit error rate approaches 1, $p_L \leq (1/G_C)\ln(1/(1-p_C))$ can be used as a tighter bound.

one logical qubit in the surface code is $2(d+1)^2$, leading to

$$Q_{\text{phys}} = 2(d+1)^2 Q_L. \tag{6.7}$$

In the case that p_{phys} is independent of the number of physical qubits, p_L can be made arbitrarily small, with cost (depending on code distance d) scaling as $d \sim \log(1/p_L)$. However, if we replace p_{phys} with the Q_{phys}-dependent function $p_{\text{phys}}(Q_{\text{phys}})$ of Eq. 6.3 (i.e. the scalability model), the logical error rates cannot be made arbitrarily small. The smallest error rate is achieved when $p_{\text{phys}} = p_{\text{th}}$, which occurs when $Q_{\text{phys}} = (p_{th}/p_0)^s$; including more qubits (i.e. increasing the code distance) will lead to a decrease in the logical error rate. This number of physical qubits is therefore the maximal number of physical qubits that should be used under the scalability model:

$$Q_{\text{phys}}^{\text{max}} = (p_{th}/p_0)^s. \tag{6.8}$$

So, for example, when $p_{th} = 0.01$ and $p_0 = 0.001$ (as is sometimes assumed for superconducting qubit resource estimates with the surface code Goings *et al.* [2022c]) we have $Q_{\text{phys}}^{\text{max}} = 10^s$. A more optimistic setting of $p_0 = 0.0001$ leads to $Q_{\text{phys}}^{\text{max}} = 10^{2s}$. Figure 6.1 depicts contours of $Q_{\text{phys}}^{\text{max}}$ in the plane of p_0/p_{th} vs s.

The above concepts can be summarized as follows:

Requirement

$p_C \geq G_C p_L$ (Algorithm Error Tolerance)

Cost

$Q_{\text{phys}} = 2(d+1)^2 Q_L$ (Surface Code Overhead)

Models

$G_C = \alpha Q_L^{\beta}$ (QPE Circuit Gate Count)

$p_L = A\left(\frac{p_{\text{phys}}}{p_{\text{th}}}\right)^{\frac{d+1}{2}}$ (Surface Code Logical Error Rate)

$p_{\text{phys}} = p_0 Q_{\text{phys}}^{1/s}$ (Scalability Physical Error Rate Model)

Putting these together, we can determine the number of physical qubits required to ensure that QPE returns an ϵ-accurate estimate (with high probability) as a function of the number of logical qubits Q_L (roughly corresponding to problem size). This relationship is expressed by Q_{phys}-Q_L pairs that ensure Eq. 6.5 is satisfied (i.e. that logical error rates are low enough for the algorithm to succeed),

$$\sqrt{8Q_L}\log\left(\frac{A\alpha}{p_C}Q_L^{\beta}\right) \leq \sqrt{Q_{\text{phys}}}\log\left(\frac{p_{\text{th}}}{p_0}Q_{\text{phys}}^{-1/s}\right). \tag{6.9}$$

Before applying this result to the quantitative example that has been set up, we make a few general remarks that apply to any algorithm analyzed in this manner.

First, we consider the right-hand side of this inequality. This function will determine an optimal value for Q_{phys}, which we label as $Q_{\text{phys}}^{\text{opt}}$. Previously, we described a maximum value of Q_{phys} as set by the condition of p_{phys} being below threshold. However, the maximum allowed value of Q_L is now set by a function of Q_{phys};

to increase this ceiling, we should maximize the right hand side function of Q_{phys}. This function achieves its maximum of $\left(\frac{2e}{s}\right)^2 \left(\frac{p_{\text{th}}}{p_0}\right)^s$ at a value of

$$Q_{\text{phys}}^{\text{opt}} = \frac{1}{e^2}\left(\frac{p_{\text{th}}}{p_0}\right)^s \leq Q_{\text{phys}}^{\text{max}}. \tag{6.10}$$

This is considered the optimal number of physical qubits in that it enables the use of the largest number of logical qubits. As an example, for $p_{\text{th}} = 0.01$, $p_0 = 0.0001$, and $s = 3.5$, the optimal number of physical qubits is $Q_{\text{phys}}^{\text{opt}} \approx 1.35 \times 10^6$.

These quantities of $Q_{\text{phys}}^{\text{max}}$ and $Q_{\text{phys}}^{\text{opt}}$ can help us to quantify the scalability parameters p_0 and s that are relevant to the NISQ-to-EFTQC transition and the EFTQC-to-FTQC transitions. At the end of the previous subsection we described how the NISQ-to-EFTQC might occur in the range of 100 to 10,000 physical qubits. Considering Equation 6.8 and 6.10, this determines the (p_0,s) pairs characteristic of this transition and shown as the red-to-green blend in Figure 6.1.

We motivate the idea that the transition from EFTQC to FTQC is characterized by how the quantum computations are “bottlenecked”. In the case of fault-tolerant quantum computing, it is envisioned that the ability to run larger and larger quantum computations is possible as long as the computations are not practically limited by resources such as time and energy. We propose that early fault-tolerant quantum computing be characterized by the regime in which the largest possible quantum computations are limited by the maximum number of physical qubits warranted in the architecture ($Q_{\text{phys}}^{\text{max}}$ or $Q_{\text{phys}}^{\text{opt}}$). Viewing time as the limiting resource, if we assume that the quantum computation must finish within a month, then this limits the problem sizes that can be accommodated accordingly. Using the quantum chemistry resource estimations of Goings *et al.* [2022c] as a point of reference, problem instances that would take a month would require on the order of 10^7 physical qubits. There may be other classes of problems that become runtime-limited when fewer or more physical qubits are required. Thus, in Figure 6.1 we depict the transition from EFTQC to FTQC as the green-to-blue gradient ranging from 10^6 to 10^8.

Second, we consider the left-hand side of Equation 6.9. Most of the parameters are contained in the factor $A\alpha/\epsilon p_C$. In Section 6.3 we will explain the importance of this factor in quantifying the "burden" placed on the elementary fault-tolerant protocols. Equation 6.9 shows that decreasing this burden factor affords a decrease in the number of physical qubits Q_{phys}. Alternatively, when fixing the number of physical qubits, a reduction in the burden factor affords an *increase* in the number of logical qubits, and subsequently the maximum problem size or "reach" of the quantum computer. The methods introduced in Section **??** will be understood to reduce this burden factor, enabling algorithms to be run using fewer physical qubits, though at the cost of an increase in runtime.

Figure 6.4 shows the contours of solutions to Equation 6.9 for several scalability values s. The most striking feature is that, for the finite values of scalability ($s < \infty$), there is a maximum-size instance (measured by Q_L) that the architecture can accommodate using the QPE algorithm. For example, in the case of $s = 3.5$, $p_0 = 0.0001$, and $p_{\text{th}} = 0.01$, we find that the largest instance that can be accommodated (i.e. the "reach" of the quantum architecture) is $Q_L \approx 90$. The maximum number of logical qubits $Q_L^{\max}$ can be solved for by setting $Q_{\text{phys}} = Q_{\text{phys}}^{\text{opt}}$in Equation 6.9 and solving for Q_L,

$$Q_L^{\max} = \frac{Q_{\text{phys}}^{\text{opt}}}{8s^2\beta^2 W\left(\sqrt{\left(\frac{A\alpha}{p_C}\right)^{\frac{1}{\beta}} \frac{Q_{\text{phys}}^{\text{opt}}}{8s^2\beta^2}}\right)^2}, \tag{6.11}$$

where $W(x)$ is the solution to $W(x)\exp(W(x)) = x$, known as the Lambert W function. Using the upper bound of $W(x) \leq \ln(x)$, we can lower bound the maximum qubit number as

$$Q_L^{\max} \geq \frac{\left(\frac{p_{\mathrm{th}}}{p_0}\right)^s}{2e^2s^2\beta^2 \ln\left(\left(\frac{A\alpha}{p_C}\right)^{\frac{1}{\beta}} \frac{\left(\frac{p_{\mathrm{th}}}{p_0}\right)^s}{8e^2s^2\beta^2}\right)^2}, \tag{6.12}$$

where we have used the expression for $Q_{\mathrm{phys}}^{\mathrm{opt}}$. This maximum solvable problem size motivates the question explored in the next section: *with a fixed scalability, is it possible to extend the "reach" of a quantum architecture using algorithms designed for finite scalability?*

6.3 Example: randomized Fourier estimation under finite scalability

Section 6.2.2 ended with the question of how we might extend the reach of finite scalability quantum computers. The previous subsection overviewed a host of quantum algorithms suited for addressing this question. In this section we take one quantum algorithm from the previous section and quantitatively investigate its ability to extend the reach of a finite scalability quantum computer for the task of phase estimation.

We use as our example the randomized Fourier estimation (RFE) algorithm as introduced in Kshirsagar *et al.* [2022] and adapted for trading circuit repetitions for number of operations per circuit in Liang *et al.* [2023b]. The RFE algorithm solves the task of phase estimation introduced in Section 6.2.2. It is an alternative to the standard quantum phase estimation (QPE) algorithm Nielsen and Chuang [2011] and related algorithms such as robust phase estimation (RPE) Kimmel *et al.* [2015].

We consider the RFE algorithm to be a prototypical quantum algorithm suited for early fault-tolerant quantum computing given that it has the following features:

- **Qubit conservation**: the (high-level) circuit conserves qubit count by using just one ancilla qubit.
- **Circuit trading**: the number of operations per circuit is tuned by input parameter K, enabling a trade-off between this quantity and the required number of circuit repetitions.
- **Robustness**: the algorithm is robust to circuit error and this robustness can be understood in terms of a signal corrupted by a noise floor.

As we will show, and like many of the other EFTQC algorithms introduced in Section **??**, these features equip the algorithm to accommodate limited scalability in the early fault-tolerant quantum computing regime. Furthermore, RFE is very simple, helping to facilitate discussion of these algorithmic concepts relevant to early fault-tolerant quantum computing.

We will a) briefly review randomized Fourier estimation, and then investigate b) how trading circuit repetitions for decrease of operations and c) how robustness to error help to increase the problem instance size (i.e. the "reach") that can be solved with a finite scalability architecture.

RFE Intro The RFE algorithm relies on the Hadamard test circuit (as depicted in Figure 6.5). Each Hadamard test circuit is parameterized by the circuit depth (k) and a phase parameter (ϕ). The output measurement probabilities correspond to an oscillatory function that encodes θ:

$$\Pr(z|\theta;k,\phi) = \frac{1}{2}(1 + z\mathrm{Re}(e^{i\phi}\langle\psi| U^k |\psi\rangle)) \tag{6.13}$$

$$= \frac{1}{2}(1 + z\cos(k\theta + \phi)) \tag{6.14}$$

$$\tag{6.15}$$

It is convenient to view the expected value of z, which is $g(k) = \cos(k\theta + \phi)$, as the true signal encoding θ. The phase θ is then

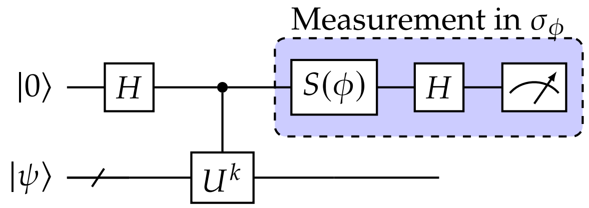


Figure 6.5 An archetypal circuit template used by many EFTQC algorithms. The measurement outcome probabilites depend on $|\psi\rangle$ and U as $\Pr(\pm 1|k,\phi) = \frac{1}{2}(1 \pm \cos(k\theta + \phi))$. Measurement outcomes can be processed to In the case of the Randomized Fourier Estimation (RFE) algorithm, the measurement outcomes encode the The parameter k is uniformly randomly chosen among $\{0,\ldots,K-1\}$ for each circuit repetition. K then controls the maximal circuit depth and is used to reduce the number of operations per circuit. The boxed-up elements in blue can be collectively interpreted as a measurement with respect to the observable $\sigma_\phi = \cos(\phi)\sigma_x - \sin(\phi)\sigma_y$, where σ_x and σ_y are the conventional Pauli operators and $S(\phi) = \begin{bmatrix} 1 & 0 \\ 0 & \exp(i\phi) \end{bmatrix}$.

estimated from measurement outcome data in a manner similar to estimating the frequency of a noisy estimate of $g(k)$. The parameters k and ϕ are chosen uniformly randomly in each sample, with $k \in [0, K-1]$ and ϕ ranging between 0 and 2π. Each measurement outcome z obtained from the circuit is used to form an unbiased estimator $\hat{f}_j = 2ze^{-i2\pi kj/J}e^{-i\phi}$ of the discrete Fourier transform of the signal $g(k)$, where J is an algorithm parameter that sets the grid size of the Fourier spectrum. The estimate of the Fourier signal can be made more accurate by taking multiple samples and averaging them:

$$\hat{f}_j = \frac{1}{M}\sum_{i=1}^{M} \hat{f}_j^{(i)}. \tag{6.16}$$

By accumulating enough measurement outcomes, one can estimate θ accurately (i.e. within ϵ) with high probability (i.e. less than $1-\delta$)

by locating the tallest peak (or point of largest magnitude) in the estimate of the discrete Fourier transform,

$$\hat{\theta} = \frac{2\pi}{J}\text{argmax}_j \frac{1}{M}\sum_{i=1}^{M} \hat{f}_j^{(i)}. \tag{6.17}$$

The algorithm's accuracy is limited by parameter J, which is set to ensure that the Fourier resolution matches the desired accuracy.

Algorithm Parameters	
J	Sets the Fourier domain grid spacing.
K	Sets the maximum number of c-U per circuit.
M	Sets the number of circuit repetitions (i.e. samples).
Error and Confidence Requirements	
$\epsilon = 2\pi/J$	Ensures that the θ-adjacent discrete frequencies are accurate.
$\delta = 8J\exp(-M/W(K,J,\lambda))$	Ensures that enough samples are taken, given K, J, and λ (see below).
Operations Per Circuit	
$\mathbb{E}G_C = \frac{K-1}{2}G_U$	Expected value, with max being $(K-1)G_U$
Circuit Repetitions	
$M = W(K,J,\lambda)\log(\frac{16\pi}{\delta\epsilon})$	Number of samples needed. See Liang *et al.* [2023b] for definition of $W(K,J,\lambda)$.

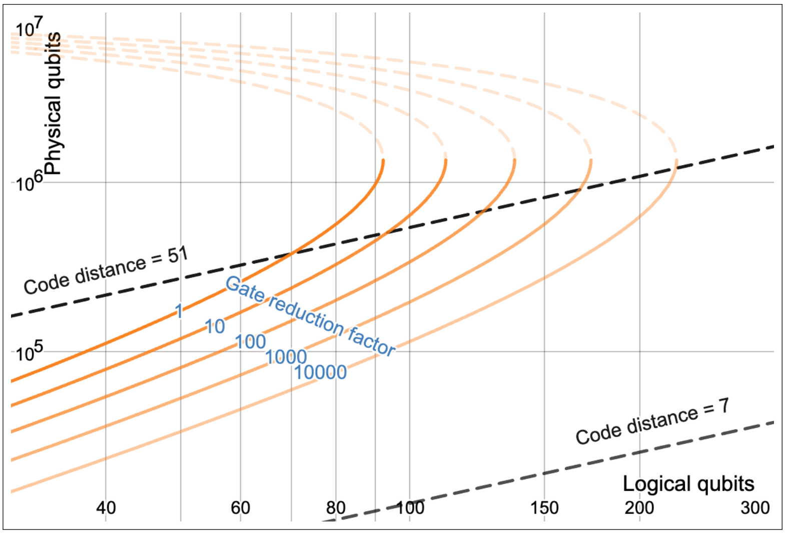


Figure 6.6 This plot shows that, under the scalability model, the EFTQC algorithm robust Fourier estimation (RFE) can extend the reach of the quantum computation from 90 logical qubits to over 200 logical qubits. This is achieved by either reducing the number of c-U used per circuit or increasing the tolerable circuit error rate p_C in the RFE algorithm. Both of these reduce the burden factor $A\alpha/\epsilon p_C$ appearing in Equation 6.9. This increase in the "reach" of the quantum computer comes at the cost of an increase in the runtime (roughly by the burden factor), which is a combination of the decrease in time per circuit and increase in number of circuit repetitions. Here we take the scalability to be $s = 3.5$ with $p_0 = 10^{-4}$, which implies that the optimal number of physical qubits is $Q^{\text{opt}}_{\text{phys}} = \frac{1}{e^2}\left(\frac{p_{\text{th}}}{p_0}\right)^s \approx 1.35 \times 10^6$. An editable version of the plot can be accessed here: https://www.desmos.com/calculator/nf43nafwet

Circuit trading We now describe how this algorithm is able to trade number of operations per circuit for circuit repetitions. The maximum number of operations per circuit (in expectation) is $(K-1)G_U$, where G_U is the number of operations in a single c-U. In the quantum phase estimation algorithm, $1/\epsilon$ calls are made to c-U, corresponding to setting $K \approx 1/\epsilon$. In RFE we can reduce the number of operations per circuit by setting K to any value less than $1/\epsilon$. This reduction in K reduces the burden factor in Equations 6.9 and 6.12 proportionally. Figure 6.6 shows how varying reductions in the burden factor lead to an increase in the problem size that RFE can accommodate. Equation 6.12 predicts that this increase in problem size grows as $O(1/\ln^2(B))$ with burden factor B. For the specific example considered, the largest problem instance can be increased from 90 to over 200 by decreasing K by a factor of 100,000.

As mentioned previously, circuit trading means that a decrease in operations per circuit comes at the cost of an increase in the number of circuit repetitions. This trade-off can be understood as follows. Decreasing K causes the *width* of the peak in the discrete Fourier spectrum to increase. With the spectrum being more flat near the peak, smaller amounts of noise in the signal are able to shift the peak location more than ϵ (leading to algorithm failure). This statistical sampling noise must then be reduced by taking more samples. The analytic relationship is given in the appendix of Liang *et al.* [2023b]. This describes the nature of the trade-off between operations per circuit and circuit repetitions.

Robustness The RFE algorithm has been analyzed in previous work with respect to three different algorithmic noise models: adversarial noise and Gaussian noise Kshirsagar *et al.* [2022] and exponential decay noise Liang *et al.* [2023b]. We give brief explanations of how the Gaussian noise and the exponential decay noise impact the algorithm performance and thus explain the robustness of the RFE algorithm to a particular model of noise. In Kshirsagar *et al.* [2022], the Gaussian noise model is analyzed, wherein it is assumed that, for each circuit (labeled by k), the output probability has been corrupted by a small perturbation drawn from a Gaussian distribution,

$$\Pr(z|\theta; k, \phi) = \frac{1}{2}(1 + z(\cos(k\theta + \phi) + \eta_k)), \tag{6.18}$$

where each η_k has been drawn from a Gaussian distribution with mean zero and standard deviation σ. How does this impact the performance of the algorithm? The η_k can be understood to corrupt the expected value of z (i.e. the signal $g(k)$). This impacts the Fourier spectrum by adding a "noise floor" related to the Fourier transform of the η_k. The algorithm can still succeed as long as this noise floor does not shift the location of the peak by more than ϵ. The authors of Kshirsagar *et al.* [2022] proved that if σ is below a certain quantity (dependent on ϵ and δ) then the algorithm can succeed with more than $1-\delta$ probability.

In Liang *et al.* [2023b], the exponential decay model is derived from a lower-level noise model. The exponential decay model assumes that the likelihood function now includes a factor that decreases exponentially in k,

$$\Pr(z|\theta; k, \phi) = \frac{1}{2}(1 + ze^{-k\lambda}\cos(k\theta + \phi)), \tag{6.19}$$

with decay parameter λ. Experiments Katabarwa *et al.* [2021]; Giurgica-Tiron *et al.* [2022a] show that this model is accurate for small systems. This exponential decay factor causes the expected value of z (i.e. the underlying signal $g(k)$) to attenuate as k is increased. In the Fourier domain, this attenuation translates into an attenuation of the peak (see this desmos plot). As with the peak broadening due to reducing K, a smaller amount of statistical noise is sufficient to shift the location of the estimated peak more than ϵ. Accordingly, more samples must be taken to sufficiently reduce this statistical noise.

Under the assumption that the exponential decay model holds exactly, Liang *et al.* [2023b] shows that with arbitrarily large decay parameter λ, the algorithm can generate an ϵ accurate estimate with probability greater than $1-\delta$. In other words, the algorithm can be made arbitrarily robust. The reason is that the exponential decay error does not shift the location of the peak in the Fourier spectrum of the expected signal. This increase in robustness translates into a decrease in the burden: allowing the circuit error rate p_C to increase towards 1 increases the allowed logical error rate p_L, decreasing the burden factor.

Consider a reduction in the burden on account of an increase in the tolerable circuit error rate p_C, which quantifies the the robustness of the algorithm. Note that in the case where p_C is close to 1, a better approximation than the union bound can be used to replace p_C with $\ln(1/(1-p_C))$, which grows to infinity as $p_C \to \infty$. We remark that in the case of the exponential decay model, the circuit error rate is $p_C = 1 - e^{-k\lambda}$, which leads to $\ln(1/(1-p_C)) = k\lambda$. Therefore, as we allow for an increase in λ, the burden factor is reduced proportionally (where we keep in mind that, for small values of p_C, the burden factor scales proportionally to it).

We previously discussed Figure 6.6 in the context of circuit trading. This figure can also be used to demonstrate the impact of increased robustness. Considering an increase in p_C to be the cause of the burden factor reduction, Figure 6.6 shows how the reach of the quantum computer is increased accordingly. As with circuit trading, there is a price paid for this extended reach of the quantum computer: for the RFE algorithm, Liang *et al.* [2023b] shows that the runtime grows exponentially in λ for $\lambda \geq 1/2$ (where K is set to its minimum value of 2). Therefore, in practice, there may be an upper limit to the degree of robustness, beyond which the runtime becomes too large to be practical. This is an issue that many error mitigation techniques face Cai *et al.* [2022]. This similarity may not be surprising in that the way RFE accommodates error is a type of error mitigation.

In practice, the exponential decay model is not exact. Instead, we expect that in any given device and compilation of c-U, the likelihood function will include some deviation (possibly varying over time) from the exponential decay model likelihoods. While in the exact exponential decay model the Fourier peak location is unchanged, allowing deviations from this model can shift the location of the peak. This sets a lower limit to the achievable accuracy ϵ, a feature which is found in the bounded adversarial noise model and the Gaussian noise model of Kshirsagar *et al.* [2022].

We have demonstrated how the RFE algorithm, as an archetypal EFTQC algorithm, enables a reduction in the burden placed on the fault-tolerant protocols. Figure 6.6 demonstrates how larger problem instance sizes can be accommodated by either reducing the number of operations per circuit (decreasing K) or by increasing the robustness of the algorithm (increasing p_C). This is because the burden factor $\frac{A\alpha}{p_C}$ incorporates both of these quantities. For both examples of reducing the burden factor, there is an increase in the runtime of the algorithm. Although the RFE algorithm enables parallelizing the circuit repetitions over multiple quantum computers to reduce runtime, the runtime is expected to be a bottleneck for many applications. Therefore, the runtime costs of reducing the fault-tolerance burden must be carefully considered. See Liang *et al.* [2023b] for an quantitative account of such runtime costs for RFE. We leave a thorough investigation of the runtime costs of decreasing the fault-tolerance burden for EFTQC algorithms to future work.

6.4 Discussion and Outlook

In this perspective we investigated the regime between NISQ and FTQC, which is referred to as "early fault-tolerant quantum computing". To understand the prospects for utility in this regime, we proposed a simple computational model to quantitatively capture the performance of quantum architectures within these three regimes. The *scalability* model characterizes the ability of a quantum hardware vendor to provide systems with low physical error rates as the requested number of physical qubits is increased. This differs from previous approaches that assume a scale-independent performance for their quantum architectures Kim *et al.* [2022a]; Goings *et al.* [2022c]; Beverland *et al.* [2022]. We demonstrated that the QPE algorithm Cleve *et al.* [1998] compiled to the surface code Fowler and Gidney [2018] has a limit on the problem size that can be accommodated by a vendor with finite scalability, according to our model. Un-surprisingly, this is due to scale-dependent error rates (Equation 6.3) combined with the diminishing returns of fault-tolerant protocols as the error rates of the device approach the numerically-estimated threshold value Wang *et al.* [2011]. Next, we showed that by using an algorithm suited to finite-scalability (the randomized Fourier estimation algorithm Kshirsagar *et al.* [2022]), when granted the same scalability, the problem size limit can be extended from around 90 qubits (for QPE) to around 130 qubits (using the same number of physical qubits). This comes at the cost of roughly a 100 times increase in runtime.

The scalability model enabled us to quantitatively discuss the transition from NISQ to EFTQC to FTQC. This transition is characterized by the waning of scale-dependent physical error rates (see Eq. 6.3). At the end of Section 6.2.1 we described how the nature of the transition from the regime of NISQ to EFTQC is difficult to predict; future advances might allow for implementing certain fault-tolerant components far sooner than current methods would enable. However, we mentioned some of the technical considerations that might govern the transition and, accordingly, depict this transition in Figure 6.1 to occur through the range of $Q_{\text{phys}}^{\text{max}}$ being 100 to 10,000. Regarding the transition from EFTQC to FTQC, we described in Section 6.2.1 how each regime might be characterized by different bottlenecks; EFTQC is characterized by the largest solvable problem instances being bottle-necked by the number of available physical qubits (or, better, $Q_{\text{phys}}^{\text{opt}}$), whereas FTQC is characterized by the largest solvable problem instances being bottle-necked by runtime. Accordingly, we explain how this transition might occur in the range of $Q_{\text{phys}}^{\text{max}}$ being 10^6 to 10^8.

Different factors, such as hardware, algorithmic, and fault-tolerance advances, play a dominant role in characterizing the EFTQC regime. The recent work of Kim *et al.* [2023a] provides evidence for the utility of noisy quantum devices in the pre-fault tolerant era and emphasizes the role of hardware advances to achieve this. Moreover, many works have highlighted the importance that quantum algorithm development has in leveraging the capabilities of the quantum devices to their maximum potential Wang *et al.* [2019, 2021]; Katabarwa *et al.* [2021]; Xiao *et al.* [2023]. Recent works have also explored the effect of noise in the performance of quantum algorithms and highlighted the need to use QEC prudently Alcazar *et al.* [2022]; Johnson *et al.* [2022]; Liang *et al.* [2023b]. This work is a first attempt to incorporate all the aforementioned factors (hardware, algorithmic, and fault-tolerance advances) in order to validate the assumption that there is a meaningful regime of early fault-tolerant quantum computing methods, which is usually assumed in papers on the subject Lin and Tong [2022a]; Wang *et al.* [2023a]; Ding and Lin [2022b]. What remains to be determined is how rapidly quantum hardware will progress through this regime; or, in other words, it remains to determine how the scalability of quantum hardware vendors will increase over time.

To put these results into context, recent resource estimates on a variety of molecules relevant to Li-on electrolyte chemistry Kim *et al.* [2022c] show that above 100 logical qubits would be necessary to tackle such systems. This indicates that extending the reach of the problem size limit from 90 logical qubits to over 130 with the framework discussed here might have interesting implications, i.e. allowing to study problems of interest before the realization of FTQC regime. Our results suggest that the EFTQC regime could exist in a meaningful way, i.e. using the same quantum resources compared to FTQC (number of physical qubits and scalability model), while affording the use of a larger number of logical qubits (Fig. 6.6).

This work explored the usefulness of the EFTQC regime for a specific quantum algorithm and QEC model, namely the RFE Kshirsagar *et al.* [2022] and the surface code Fowler and Gidney [2018]. The underlying methodology, however, can be easily extended to other algorithms and fault-tolerant protocols while using the suggested or alternative scalability models. Although the proposed model of scalability is quite general, we do not expect it to perfectly fit the scalability profile of vendors over many orders of magnitude. But, we anticipate that it could capture the qualitative behavior over at least a few orders of magnitude. Moreover, we showed in Appendix F.2 that, even when a more optimistic model is used (specifically a logarithmic model), the qualitative finding remains: there is an upper limit on the size of the quantum computation.

Future work could explore other models of scalability that, for example, might be given directly by the hardware provider and accommodate the features of the architecture as it is scaled. Another interesting direction is to adapt the scalability model to address the interplay between quantum error mitigation and quantum error correction Piveteau *et al.* [2021]; Suzuki *et al.* [2022] which will help drive the transition from NISQ to EFTQC. Moreover, the proposed framework could be applied to other combinations of algorithms and quantum error correcting codes and be used to examine the utility of the EFTQC regime for other potential application fields of quantum computing.

Our work provides evidence for the utility of the EFTQC regime within a framework that includes crucial factors of quantum computing, such as hardware, algorithm, and fault-tolerance advances. To incorporate the hardware advances, we have introduced a simple scalability model to capture the performance of devices that are continually improving. As it is yet unclear how exactly quantum devices will scale up to incorporate millions or billions of physical qubits Preskill [2018], the proposed model of scalability is just a first attempt to bridge the gap between NISQ and FTQC. Future works in these directions could help move beyond the NISQ-FTQC dichotomy and further explore how EFTQC might deliver practical quantum advantage at scale.

6.5 Conlcusions

In this perspective work we explored the regime between NISQ and FTQC referred to as early fault-tolerant quantum computing

(EFTQC). This work is a first attempt to incorporate hardware, algorithmic and fault-toletant advances to validate that EFTQC will exist in a meaningful way. Next, it would be interesting to apply a similar analysis to other application areas of quantum computing.

Chapter 7

Conclusions

This thesis focused on introducing tools to quantify the performance of quantum computing algorithms and their applications. The main focus was on the two most popular application areas of quantum computing, quantum machine learning (QML) and quantum chemistry.

7.1 Quantum machine learning

Following the methods used for the early development of neural networks, namely statistical physics techniques of spin glasses we analyzed the storage properties of quantum perceptrons models with a direct hardware implementation. This work merged quantum information with statistical physics (referred to as Gardner's program) and opened a new approach of studying quantum models and their learning capabilities. Next, we applied the same methodology to quantum neural networks where once again one crucial feature is their storage capacity for associative memory, that is, the number of patterns (stored memories/attractors) the network has for a given number of neurons. These tools could help us build our fundamental understanding of quantum machine learning models and their capabilities.

Eventually, like with classical machine learning models large-scale computational benchmarking will be necessary to understand the maximum potential of quantum machine learning models. To this end, we introduced the teacher-student scheme as a computational tool to benchmark the performance of quantum models. With the teacher-student scheme, we can benchmark two or more different QML models directly, without the use of any artificial data that might unwittingly favor the performance of one model versus the others. Importantly, the teacher-student scheme could help explore a potential quantum advantage of certain models in terms of their training capabilities.

Until large-scale benchmarking is available, small-scale quantum simulations could help better understand the role of different elements in quantum models. Through numerical simulations on small size systems, we explored how ancilla qubits could help improve and how the measurement location could affect the overall the performance of quantum models. This work brought together different tools discussed in the literature to explore the training capabilities of quantum models for quantum machine learning applications.

7.2 Quantum Chemistry

In recent years substantial research effort has been devoted to quantum algorithms for ground state energy estimation (GSEE) in chemistry and materials. For most GSEE algorithms, the runtime depends on the ground state preparation (GSP) method. Initially, we introduce the efficiency criteria which provide a method to carefully assessing the utility of various combinations of GSEE and GSP methods, by accepting or rejecting a GSP method for the purposes of GSEE. This work sets a foundation from which to further explore the requirements to achieve quantum advantage in quantum chemistry.

Next, we used the efficiency criteria for extensive numerical simulations to investigate whether in those cases quantum heuristic GSP methods could improve the overlap values compared to Hartree-Fock, which is one of the most common method in classical chemistry. Our findings indicate that quantum heuristic GSP can accelerate GSEE tasks, already for computationally affordable strongly-correlated systems of intermediate size. These results suggest that quantum heuristic GSP has the potential to significantly reduce the runtime requirements of GSEE algorithms, thereby enhancing their suitability for implementation on quantum hardware.

Finally, over the past decade, research in quantum computing has tended to fall into one of two camps: near-term intermediate scale quantum (NISQ) and fault-tolerant quantum computing (FTQC). Focusing again on the task of energy estimation, we explored how to use quantum computers in transition between these two eras, namely the early fault-tolerant quantum computing (EFTQC) regime. This work was a first attempt to incorporate hardware, algorithmic and fault-toletant advances to validate that EFTQC will exist in a meaningful way.

Appendices

Appendix A

A.1 Abstracts of research papers

A.1.1 Storage properties of a quantum perceptron

Driven by growing computational power and algorithmic developments, machine learning methods have become valuable tools for analyzing vast amounts of data. Simultaneously, the fast technological progress of quantum information processing suggests employing quantum hardware for machine learning purposes. Recent works discuss different architectures of quantum perceptrons, but the abilities of such quantum devices remain debated. Here, we investigate the storage capacity of a particular quantum perceptron architecture by using statistical mechanics techniques and connect our analysis to the theory of classical spin glasses. Specifically, we focus on one concrete quantum perceptron model and explore its storage properties in the limit of a large number of inputs.

A.1.2 Storage capacity and learning capability of quantum neural networks

We study the storage capacity of quantum neural networks (QNNs), described by completely positive trace preserving (CPTP) maps acting on a N-dimensional Hilbert space. We demonstrate that attractor

QNNs can store in a non-trivial manner up to N linearly independent pure states. For n qubits, QNNs can reach an exponential storage capacity, $\mathcal{O}(2^n)$, clearly outperforming standard classical neural networks whose storage capacity scales linearly with the number of neurons n. We estimate, employing the Gardner program, the relative volume of CPTP maps with $M \leq N$ stationary states and show that this volume decreases exponentially with M and shrinks to zero for $M \geq N + 1$. We generalize our results to QNNs storing mixed states as well as input-output relations for feed-forward QNNs. Our approach opens the path to relate storage properties of QNNs to the quantum features of the input-output states.

A.1.3 Exploring quantum perceptron and quantum neural network structures with a teacher-student scheme

Near-term quantum devices can be used to build quantum machine learning models, such as quantum kernel methods and quantum neural networks (QNN) to perform classification tasks. There have been many proposals on how to use variational quantum circuits as quantum perceptrons or as QNNs. The aim of this work is to introduce a teacher-student scheme that could systematically compare any QNN architectures and evaluate their relative expressive power. Specifically, the teacher model generates the datasets mapping random inputs to outputs which then have to be learned by the student models. This way, we avoid training on arbitrary data sets and allow to compare the learning capacity of different models directly via the loss, the prediction map, the accuracy and the relative entropy between the prediction maps. Here, we focus particularly on a quantum perceptron model inspired by the recent work of Tacchino et. al. Tacchino *et al.* [2019] and compare it to the data re-uploading scheme that was originally introduced by Pérez-Salinas et. al. Pérez-Salinas *et al.* [2020]. We discuss alterations of the perceptron model and the formation of deep QNN to better understand the role of hidden units and the non-linearities in these architectures.

A.1.4 The effect of the processing and measurement operators on the expressive power of quantum models

There is an increasing interest in Quantum Machine Learning (QML) models, how they work and for which applications they could be useful. There have been many different proposals on how classical data can be encoded and what circuit ansätze and measurement operators should be used to process the encoded data and measure the output state of an ansatz. The choice of the aforementioned operators plays a determinant role in the expressive power of the QML model. In this work we investigate how certain changes in the circuit structure change this expressivity. We introduce both numerical and analytical tools to explore the effect that these operators have in the overall performance of the QML model. These tools are based on previous work on the teacher-student scheme, the partial Fourier series and the averaged operator size. We focus our analysis on simple QML models with two and three qubits and observe that increasing the number of parameterized and entangling gates leads to a more expressive model for certain circuit structures. Also, on which qubit the measurement is performed affects the type of functions that QML models could learn. This work sketches the determinant role that the processing and measurement operators have on the expressive power of simple quantum circuits.

A.1.5 Evaluating the efficiency of ground state preparation algorithms

In recent years substantial research effort has been devoted to quantum algorithms for ground state energy estimation (GSEE) in chemistry and materials. Given the many heuristic and non-heuristic methods being developed, it is challenging to assess what combination of these will ultimately be used in practice. One important metric for assessing utility is runtime. For most GSEE algorithms, the runtime depends on the ground state preparation (GSP) method. Towards assessing the utility of various combinations of GSEE and GSP methods, we asked under which conditions a GSP method should be accepted over a reference method, such as the Hartree-Fock state. We introduce a criteria for accepting or rejecting a GSP method for the purposes of GSEE. We consider different GSP methods ranging from heuristics to algorithms with provable performance guarantees and perform numerical simulations to benchmark their performance on different chemical systems, starting from small molecules like the hydrogen atom to larger systems like the jellium. In the future this approach may be used to abandon certain VQE ansatzes and other heursitics. Yet so far our findings do not provide evidence against the use of VQE and more expensive heuristic methods, like the low-depth booster. This work sets a foundation from which to further explore the requirements to achieve quantum advantage in quantum chemistry.

A.1.6 Early-fault tolerant quantum computing

Over the past decade, research in quantum computing has tended to fall into one of two camps: near-term intermediate scale quantum (NISQ) and fault-tolerant quantum computing (FTQC). Yet, a growing body of work has been investigating how to use quantum computers in transition between these two eras. This envisions operating with tens of thousands to millions of physical qubits, able to support fault-tolerant protocols, though operating close to the fault-tolerant threshold. Two challenges emerge from this picture: how to model the performance of devices that are continually improving and how to design algorithms to make the most use of these devices? In this work we develop a model for the performance of early fault-tolerant quantum computing (EFTQC) architectures and use this model to elucidate the regimes in which algorithms suited to such architectures are advantageous. As a concrete example, we show that, for the canonical task of phase estimation, in a regime of moderate scalability and using just over one million physical qubits, the "reach" of the quantum computer can be extended (compared to the standard approach) from 90-qubit instances to over 130-qubit instances using a simple early fault-tolerant quantum algorithm, which reduces the number of operations per circuit by a factor of 100 and increases the number of circuit repetitions by a factor of 10,000. This clarifies the role that such algorithms might play in the era of limited-scalability quantum computing.

A.1.7 Comparing Classical and Quantum Ground State Preparation Heuristics

One promising field of quantum computation is the simulation of quantum systems, and specifically, the task of ground state energy estimation (GSEE). Ground state preparation (GSP) is a crucial component in GSEE algorithms, and classical methods like Hartree-Fock state preparation are commonly used. However, the efficiency of such classical methods diminishes exponentially with increasing system size in certain cases. In this study, we investigated whether in those cases quantum heuristic GSP methods could improve the overlap values compared to Hartree-Fock. Moreover, we carefully studied the performance gain for GSEE algorithms by exploring the trade-off between the overlap improvement and the associated resource cost in terms of T-gates of the GSP algorithm. Our findings indicate that quantum heuristic GSP can accelerate GSEE tasks, already for computationally affordable strongly-correlated systems of intermediate size. These results suggest that quantum heuristic GSP has the potential to significantly reduce the runtime requirements of GSEE algorithms, thereby enhancing their suitability for implementation on quantum hardware.

Appendix B

B.1 Methods

In this section, we elaborate on the computational details presented in Sec. 2.2.5, i.e., the averaging over the inputs, the calculation of the effective potential, and the saddle-point approximation.

B.1.1 Averaging over the input patterns

We perform the average $\langle\langle\cdot\rangle\rangle$ and assume weak correlations between the weights. Then we can approximate

$$\langle\langle\prod_{\alpha,\mu} e^{-\frac{i}{m}x_\mu^\alpha|\vec{i}^\mu\cdot\vec{w}^\alpha|^2}\rangle\rangle = e^{\frac{-i}{m}\sum_\alpha x_\mu^\kappa \sum_\kappa(\omega_\kappa^\alpha)}\times$$

$$\prod_{\mu,k\neq l}\cos\left(\frac{2}{m}\sum_\alpha x_\mu^\alpha w_k^\alpha w_l^\alpha\right). \tag{B.1}$$

The above expressions are somewhat similar to those derived in Kohring [1990], but not the exponential phase factor, effectively shifting the values of λ's by -1, and the pre-factor $2/m$, multiplying the argument of the cos(.) function. Using Eq. (B.1) the relative volume becomes

$$\langle\langle V_S \rangle\rangle = \frac{1}{V_{S_0}^n} \int_w \int_\lambda \int_x \int_E \int_q \int_F$$
$$\times \exp\left[i \sum_{\alpha,\mu} x_\mu^\alpha \left(\lambda_\mu^\alpha - 1 \right) + \sum_{\mu,k,l} \log\cos\left(\tfrac{2}{m} \sum_\alpha x_\mu^\alpha w_k^\alpha w_l^\alpha \right) \right]$$
$$\times \exp\left[iE^\alpha \left(|\vec{w}^\alpha|^2 - m \right) + imF^{\alpha\beta}(q^{\alpha\beta} - \frac{1}{m} \sum_k w_k^\alpha w_k^\beta) \right] . \qquad \text{(B.2)}$$

Next, we use the approximation $\log\cos x \approx -x^2/2$ and use Eq. (2.11). In addition, we employ that the integral

$$I = \int_{-\infty}^{\infty} \prod_{\alpha,\mu} \frac{dx_\mu^\alpha}{2\pi} \int_\kappa^\infty \prod_{\alpha,\mu} d\lambda_\mu^\alpha$$
$$\times \exp\left(i \sum_{\alpha,\mu} x_\mu^\alpha \left(\lambda_\mu^\alpha - 1 \right) - \sum_{\mu,\alpha} \left(x_\mu^\alpha \right)^2 - 2 \sum_{\alpha<\beta,\mu} (q^{\alpha\beta})^2 x_\mu^\alpha x_\mu^\beta \right) \qquad \text{(B.3)}$$

factorizes according to

$$I = \left[\int_{-\infty}^{\infty} \prod_\alpha \frac{dx^\alpha}{2\pi} \int_\kappa^\infty \prod_\alpha d\lambda^\alpha \right.$$
$$\left. \times\, e^{i \sum_\alpha x^\alpha (\lambda^\alpha - 1) - \sum_\alpha (x^\alpha)^2 - 2 \sum_{\alpha<\beta} (q^{\alpha\beta})^2 x^\alpha x^\beta} \right]^p , \qquad \text{(B.4)}$$

which leads to Eq. (2.12) - (2.15).

B.1.2 Calculation of G_1

We assume replica symmetry of $q^{\alpha\beta}$ and after the integration over x^μ we have

$$\lim_{n\to 0} \frac{1}{n} G_1[q] = \int_{-\infty}^{\infty} Dy \log L(y) , \qquad \text{(B.5)}$$

where we used the abbreviation Eq. (2.31) and introduce

$$L(y) = 2\sqrt{\pi}\,\mathrm{Erfc}\left[\frac{\kappa - 1 + yq}{\sqrt{2(1-q^2)}}\right]. \tag{B.6}$$

The function $L(y)$ is the main object that distinguishes the classical and the quantum perceptron. In the classical case we have

$$L(y) = 2\sqrt{\pi}\,\mathrm{Erfc}\left[\frac{\kappa + yq}{\sqrt{(1-q)}}\right]. \tag{B.7}$$

In the quantum case $L(y)$ depends on q^2, since we are dealing with squared scalar products, which leads to an additional factor of two in the denominator of $L(y)$; this factor will then be responsible for the increase of the storage capacity for the quantum case in comparison to the classical case.

B.1.3 Calculation of G_2 for spherical weights

We also assume replica symmetry of E^α and $F^{\alpha\beta}$ and perform the multi-dimensional Gaussian integral in Eq. (2.15) resulting in

$$G_2[E,F] = \log\left[(2\pi i)^{n/2}(\det M)^{-1/2}\right], \tag{B.8}$$

where we introduced the matrix

$$M^{ab} = (2E+F)\delta^{ab} - F\,. \tag{B.9}$$

The matrix M has $n-1$ degenerate eigenvalues $\Lambda_1 = \ldots = \Lambda_{n-1} = 2E + F$ and one non-degenerate eigenvalue $\Lambda_n = 2E - (n-1)F$ such that the determinant of the matrix M becomes

$$\log\det M = (n-1)\log(2E+F) + \log[2E-(n-1)F]\,. \tag{B.10}$$

B.1.4 Saddle-point equations of G for spherical weights

Since G_1 does not depend on E and F the saddle-point equations with respect to E and F are

$$0 = \frac{1}{n}\frac{\partial G}{\partial E} = -i + \frac{1}{n}\frac{\partial G_2}{\partial E}, \tag{B.11a}$$

$$0 = \frac{1}{n}\frac{\partial G}{\partial F} = \frac{i}{2}(n-1)q + \frac{1}{n}\frac{\partial G_2}{\partial F}, \tag{B.11b}$$

with

$$\frac{1}{n}\frac{\partial G_2}{\partial E} = \frac{(n-1)F + 2E(n-2)}{2(2E+F)(-Fn+2E+F)}, \tag{B.12a}$$

$$\frac{1}{n}\frac{\partial G_2}{\partial F} = \frac{(n-1)F}{2(2E+F)(-Fn+2E+F)}. \tag{B.12b}$$

Performing the limit $n \to 0$ and solving for E and F results in

$$E = \frac{i(1-2q)}{2(1-q)^2}, \tag{B.13a}$$

$$F = \frac{iq}{(1-q)^2}. \tag{B.13b}$$

Further, we define the effective potential

$$g = \lim_{n\to 0}\frac{1}{n}G \tag{B.14}$$

and insert the solution of the saddle point equation into G. As a result, we obtain

$$g = \alpha \int_{-\infty}^{\infty} Dy \log L(y) + \frac{1}{2}\log(1-q) + \frac{1}{2(1-q)} \tag{B.15}$$

plus constant terms independent of q. We can interpret the averaged logarithm of the volume, g, as a kind of free energy (an effective potential), which is a regular function of $0 \leq q < 1$, but has a singularity at $q = 1$. To analyze the role of this singularity, we employ the asymptotic expansion of $\mathrm{Erfc}(x) \approx \sqrt{\pi} x^{-1} e^{-x^2} \theta(x)$ for $x \to \infty$. We see that

$$g \simeq -\frac{\alpha}{2(1-q)} \int_{1-\kappa}^{\infty} Dy(\kappa - 1 + y)^2 + \frac{1}{2\,(1-q)}\,, \tag{B.16}$$

has two singular terms as $q \to 1$.

We observe that there is a here is phase transition. When α is small the term $1/2(1-q)$ is dominant at q going to 1, and "pushes" the minimum of g to zero. For $\alpha > \alpha_c(\kappa)$, where

$$\alpha_c(\kappa) = [\int_{1-\kappa}^{\infty} Dy(\kappa - 1 + y)^2]^{-1}\,, \tag{B.17}$$

the term with $1/(1-q)$ is negative, and the minimum of g is at $-\infty$ so that in effect the relative volume shrinks to zero. One can check explicitly that $\alpha_c(\kappa)$ is a decreasing function of κ. The free energy at $q = 0$ becomes -

$$g(q = 0; \kappa) = \alpha \log[2\sqrt{\pi}\,\mathrm{Erfc}\left[\frac{\kappa - 1}{\sqrt{2}}\right]] + \frac{1}{2}\,, \tag{B.18}$$

or after normalization

$$\tilde{g}(q = 0) = g(q = 0; \kappa) - g(q = 0; \kappa = 0)\,. \tag{B.19}$$

We plot the effective potential as a function of α and κ in Fig. B.1.

The above analysis implies that the critical value of the storage capacity for $\kappa = 1$, $\alpha_c. = 2$. Since $\alpha_c(\kappa)$ grows as κ becomes smaller, performing the integral leads to the maximal critical storage capacity of $\alpha_{c,\max} > 2$ for $\kappa \to 0$.

B.1.5 Calculation of G_1 for Gaussian distributed inputs

First, we define the matrix

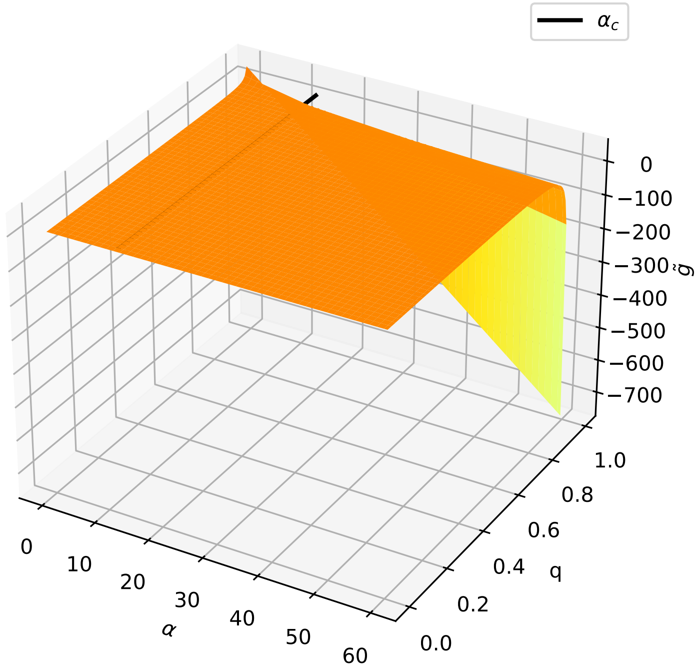


Figure B.1 The effective potential for the spherical weights. We plot the normalized effective potential $\tilde{g}$ of Eq. (B.19). It changes from finite negative values below α_c to $-\infty$ (represented by the grey area) above α_c. The black line shows the α_c curve given by Eq. (B.36).

$$A_{kk'} = \frac{1}{n} \sum_{\alpha} x^{\alpha} w_{k}^{\alpha} w_{k'}^{\alpha} \,, \tag{B.20}$$

which is spanned by the vectors w^{α} We can write its nontrivial eigenvectors as combinations of w^{α}. The eigenvalue equation is

$$\frac{1}{n} \sum_{\alpha,k'} x^{\alpha} w_{k}^{\alpha} w_{k'}^{\alpha} \sum_{\beta} c_{\beta} w_{k'}^{\beta} = \Lambda \sum_{\alpha} c_{\alpha} w_{k}^{\alpha} \,. \tag{B.21}$$

All other eigenvectors of $1 + 2i\hat{A}$ (orthogonal to the vectors w^{α}) are trivial – they correspond to eigenvalues 1 and do not contribute to the log det. Comparing coefficients, using the definition of $q^{\alpha\beta}$ and assuming replica symmetry $q^{\alpha\beta} = q$ for $\alpha \neq \beta$, leads to a closed equation

$$1 = \sum_{a} \frac{x^{\alpha} q}{\Lambda - x^{\alpha}(1 - q)} \tag{B.22}$$

for eigenvalues Λ of $\hat{A}$. Using the eigenvalues of $\hat{A}$ we can rewrite G_1 as

$$G_1[q] = \log \int \prod_{\alpha} d\lambda^{\alpha} \int \prod_{\alpha} dx^{\alpha} \frac{\exp\left(i \sum_{\alpha} x^{\alpha} \lambda^{\alpha}\right)}{\prod_{n} \left(1 + 2i\Lambda_{n}\right)}. \tag{B.23}$$

In order to rewrite the product of eigenvalues we transform the self-consistent equation for the eigenvalues Λ into the characteristic polynomial of $\hat{A}$ to define the function

$$\begin{aligned} W(\Lambda, x) = & \prod_{\alpha} [\Lambda - (1-q)x^{\alpha}] \\ & - \sum_{\alpha} q x^{\alpha} \prod_{\alpha \neq \beta} \left[\Lambda - (1-q)x^{\beta}\right] \,. \end{aligned} \tag{B.24}$$

Next, we introduce the auxiliary quantity

$$L(\epsilon) = \log\det(1 + 2i\epsilon\hat{A}) = \sum_n \log(1 + 2i\epsilon\Lambda_n)\,, \tag{B.25}$$

where we are interested in the value of $L(1)$. Differentiating with respect to ϵ we obtain

$$\frac{dL}{d\epsilon} = \frac{n}{\epsilon} - \frac{1}{\epsilon}\sum_\Lambda \frac{1}{1 + 2i\Lambda\epsilon}\,. \tag{B.26}$$

The sum can be rewritten by using Cauchy's theorem and employing an appropriate contour C. Using this integral representation for the sum we obtain

$$\frac{dL}{d\epsilon} = \frac{n}{\epsilon} + \frac{d}{d\epsilon}\log W\left(i/(2\epsilon), x\right)\,. \tag{B.27}$$

Integrating ϵ from 0 to 1 we get

$$L(1) = \log W\left(i/2, x\right) \tag{B.28}$$

since $L(\epsilon)$ goes to zero for $\epsilon \longrightarrow 0$. Expanding Λ in q, i.e., treating q as a perturbation, G_1 becomes

$$\begin{aligned} G_1[q] = \log &\int \prod_\alpha d\lambda^\alpha \int \prod_\alpha dx^\alpha \exp\left(i\sum_\alpha x^\alpha\lambda^\alpha\right) \\ &\times \frac{1}{\prod_\alpha\left(\frac{i}{2} - x_\alpha\right)}\left[1 + \sum_{\alpha,\beta}\frac{q^2 x^\alpha x^\beta}{\left(\frac{i}{2} - x_\alpha\right)\left(\frac{i}{2} - x_\beta\right)}\right]. \end{aligned} \tag{B.29}$$

Performing the integration over x^α and λ^α gives

$$G_1(q) = \frac{\kappa n}{2} - \frac{1}{2}nq^2\left(2 + \kappa\right)^2. \tag{B.30}$$

Due to the perturbative expansion in q the function $G_1(q)$ does not exhibit any singularity at $q = 1$, which will affect the nature of the phase transition, as we will see below.

B.1.6 Saddle-point equations of G for Gaussian distributed inputs

The effective potential of Eq. (2.13) becomes

$$g = \alpha\left[-\frac{\kappa}{2} - \frac{1}{2}q^2(2+\kappa)^2\right] + \frac{1}{2}\log(1-q) + \frac{1}{2(1-q)}, \quad \text{(B.31)}$$

plus constant terms independent of q. This function does contain a singular term $(1-q)^{-1}$, which repulses the saddle point solutions for the minimal value away from $q = 1$. Indeed, taking the derivative of g with respect to q gives

$$\alpha\,(2+\kappa)^2\, q = \frac{q}{2\,(1-q)^2}. \quad \text{(B.32)}$$

This equation has a trivial solution $q = 0$ for which g becomes minimal, and

$$\tilde{g} = g(\kappa) - g(\kappa = 0) = -(\alpha\kappa)/2 \quad \text{(B.33)}$$

This equation has also a non-trivial solution, which exists for $2\alpha\,(2+\kappa)^2 \geq 1$. The critical value of the storage capacity is given by (see App. B.5)

$$\alpha_c = \frac{1}{2(2+\kappa)^2}. \quad \text{(B.34)}$$

Note that the phase transition, in this case, has a different character: for both solutions, g, or more importantly $\tilde{g}$, take finite negative values, but it changes from $-(\alpha\kappa)/2$ in the "easy to learn" phase to larger negative values in the "hard to learn phase", see Fig. B.2. This behavior might be the result of expansion in q that we used to obtain the effective potential. In the "easy to learn" phase, the relative volume is 1 at $\kappa = 0$, as expected, but decreases moderately slowly exponentially with m as $\exp(m\tilde{g})$ for non-zero κ. This exponential decrease becomes much faster in the "hard to learn phase", as illustrated in Fig. (B.2). Furthermore, the effective potential might be the first term of expansion of $(1-q^2)^{-1}$ and we discuss this idea in App. B.5, which will bring us back to the volume shrinking phase transition à la Gardner.

B.1.7 *Calculation of G_2 for Ising weights*

Assuming replica symmetry of F^{ab} and averaging over the binary weights, Eq. (2.27) becomes

$$G_2 = -\frac{1}{2}Fn + n\int_{-\infty}^{\infty} Dz \log\left[2\cosh(z\sqrt{F})\right], \tag{B.35}$$

as in the classical case Krauth and Mézard [1989].

B.1.8 *Saddle-point equations of G for Ising weights*

The effective potential with replica symmetry of q^{ab} and F^{ab} becomes

$$g = \alpha\int_{-\infty}^{\infty} Dy \log L(y) + R(F,q) \tag{B.36}$$

with

$$R(F,q) = -\frac{1}{2}F(1-q) + \int_{-\infty}^{\infty} Dz \log\left[2\cosh\left(z\sqrt{F}\right)\right]. \tag{B.37}$$

Then, the saddle-point equation with respect to F is

$$-\frac{1}{2}(1-q) + \int_{-\infty}^{\infty} Dz \frac{z}{2\sqrt{F}} \tanh\left(z\sqrt{F}\right) = 0, \tag{B.38}$$

which is similar to equation obtained for the classical perceptron with binary weights Gardner and Derrida [1988], where it was argued that the solution with $F \to \infty$ as $q \to 1$ is invalid. Instead, for the correct solution of the classical perceptron problem, replica-symmetry breaking must be taken into account.

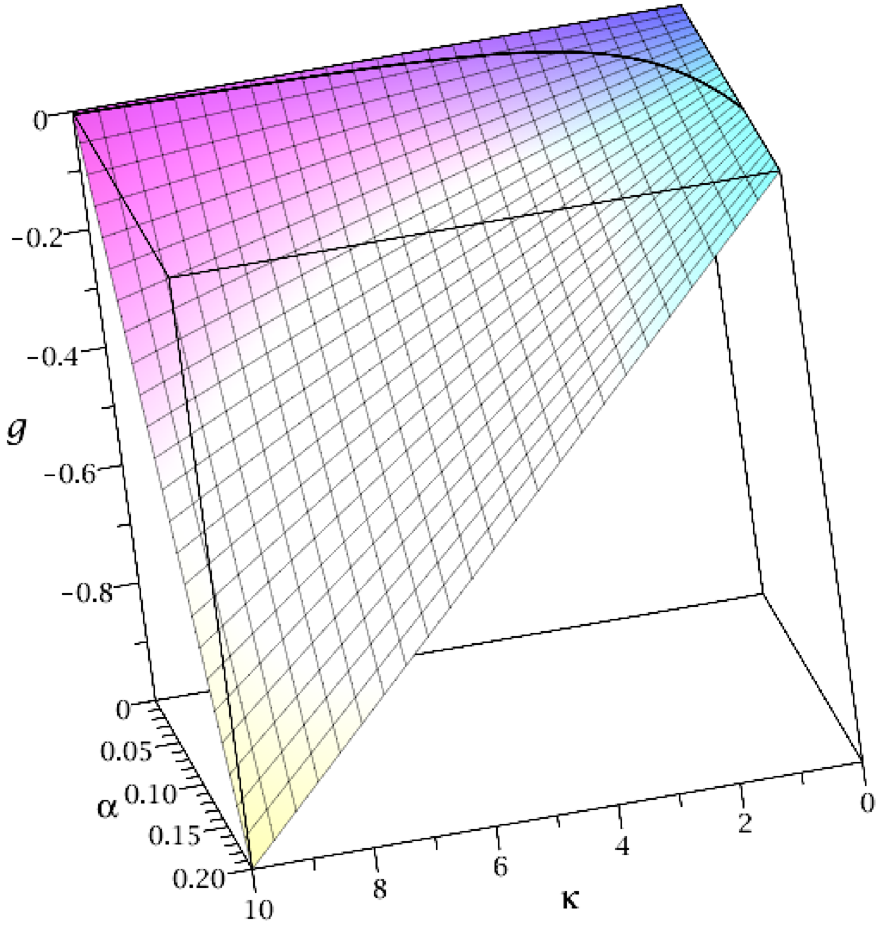


Figure B.2 The effective potential for Gaussian distributed inputs. We plot the effective potential g at the minimum, i.e. at $q = 0$ for $\alpha \leq \alpha_c$, and $q = 1 - \sqrt{1/[2(2+\kappa)^2\alpha]}$ for $\alpha > \alpha_c$. It changes from small negative values close to zero of Eq. (B.33) below the α_c curve to larger negative values of Eq. (B.31) above. The black line shows the α_c curve given by Eq. (B.34).

Here, we analyze results for replica symmetric case and compare them with Monte Carlo simulations. For the quantum perceptron, F is a well defined function of q, and it tends to infinity as $q \rightarrow 1$. The last equation can be solved

$$\sqrt{F} \approx \sqrt{\frac{2}{\pi}} \frac{1}{(1-q)}, \tag{B.39}$$

where sign$(x) = x/|x|$ and we approximated the tanh by sign(x).

Comparing the leading terms when $q \rightarrow 1$, we arrive at

$$g \approx \frac{1}{(1-q)} \left[-\frac{\alpha}{4} \int_{-\kappa}^{\infty} Dy(\kappa + y)^2 + \frac{1}{\pi} \right]. \tag{B.40}$$

In this way we obtain the critical value

$$\alpha_c(\kappa) = \frac{4}{\pi} \left[\int_{-\kappa}^{\infty} Dy(\kappa + y)^2 \right]^{-1}, \tag{B.41}$$

which implies the maximal value of $\alpha_c(0) = 8/\pi$. In contrast the MC simulations suggest that $\alpha_c(0) \simeq 0$ as illustrated in Fig. D.1. As in the classical case, we interpret this discrepancy as the necessity of replica symmetry breaking.

B.2 Details on the quantum perceptron proposed in Tacchino *et al.* [2019]

The first unitary $U_{\vec{i}}$ should fulfill

$$|\psi_{\vec{i}}\rangle = U_{\vec{i}} |0\rangle^{\otimes N}, \tag{B.42}$$

and in this way encodes the information on N qubits. Particularly, any $m \times m$ unitary matrix with the first column being identical with

$\vec{i}$ and normalized is a valid candidate for such a unitary. The information is processed by applying the second unitary $V_{\vec{w}}$ which fulfills

$$V_{\vec{w}} \left|\psi_{\vec{w}}\right\rangle = |1\rangle^{\otimes N} = |m-1\rangle \, . \tag{B.43}$$

Applying the unitary $V_{\vec{w}}$ on the encoded state leads to

$$\left|\phi_{\vec{i},\vec{w}}\right\rangle \equiv V_{\vec{w}} \left|\psi_{\vec{i}}\right\rangle = \sum_{j=0}^{m-1} c_j \, |j\rangle \; . \tag{B.44}$$

Performing multi-controlled NOT gates with a readout qubit leads to the state

$$|\phi_{i,w}\rangle \, |0\rangle = \sum_{j=0}^{m-2} c_j \, |j\rangle \, |0\rangle + c_{m-1} \, |m-1\rangle \, |1\rangle \, . \tag{B.45}$$

As a result, when measuring 1 on the readout qubit, the probability amplitude is

$$|c_{m-1}|^2 = |\vec{i}^{\mu} \cdot \vec{w}|^2. \tag{B.46}$$

B.3 Abbreviations

In this appendix, we summarize the abbreviation used in the main text. In Eq. (2.5) and Eq. (2.6) we used

$$\int_w = \int_{-\infty}^{\infty} \prod_k dw_k \, , \tag{B.47}$$

and in Eq. (2.9) the measure is

$$\begin{aligned} \int_w \int_\lambda \int_x \int_E = & \int_{-\infty}^{\infty} \prod_k dw_k \int_\kappa^{\infty} \prod_\mu d\lambda^\mu \\ & \times \int_{-\infty}^{\infty} \prod_\mu \frac{dx^\mu}{2\pi} \int_{-\infty}^{\infty} \frac{dE}{2\pi} \, , \end{aligned} \tag{B.48}$$

in Eq. (2.12) the abbreviation means

$$\int_F \int_q \int_E = \int_{-\infty}^{\infty} \prod_{\alpha<\beta} dq^{\alpha\beta} \int_{-\infty}^{\infty} \prod_{\alpha<\beta} \frac{dF^{\alpha\beta}}{2\pi} \int_{-\infty}^{\infty} \prod_{\alpha} \frac{dE^{\alpha}}{2\pi}, \tag{B.49}$$

and in Eq. (B.2) we used

$$\begin{aligned} \int_w \int_\lambda \int_x \int_E \int_q \int_F = &\int_{-\infty}^{\infty} \prod_{k,\alpha} dw_k^\alpha \int_\kappa^\infty \prod_{\mu,\alpha} d\lambda_\mu^\alpha \int_{-\infty}^{\infty} \prod_{\alpha,\mu} \frac{dx_\mu^\alpha}{2\pi} \\ &\times \int_{-\infty}^{\infty} \prod_{\alpha} \frac{dE^\alpha}{2\pi} \int_{-\infty}^{\infty} \prod_{\alpha<\beta} dq^{\alpha\beta} \int_{-\infty}^{\infty} \prod_{\alpha<\beta} \frac{dF^{\alpha\beta}}{2\pi}. \end{aligned} \tag{B.50}$$

B.4 Monte Carlo simulation

We apply the Monte Carlo simulation of the classical perceptron with Ising weights Gardner and Derrida [1989b] to the quantum perceptron. Here, we elaborate the details of the Monte Carlo simulation.

The first pattern $i_j^1 = \pm 1$ is chosen at random and we fix a certain threshold κ. Then, we go through all possible realizations of the weights and keep only the weights that satisfy the given threshold κ. This forms the remaining set of the weights. Then, a second pattern is chosen at random and we go through all possible realizations of the remaining set of the weights to keep again only the subset of weights which satisfy the given threshold κ. Then, we continue by choosing more random patterns and updating the set of the weights that fulfill the given threshold. After a certain number of P patterns that have been introduced to the perceptron, no choice for the weights exist for $P + 1$ patterns. This means that for this sample the system can store exactly P patterns.

Therefore, the value of P depends on the random choices of the $\sum i_j^\mu = \pm 1$ and the threshold κ. The threshold κ in the classical case is zero, since the output is either ±1. For the quantum case, the threshold κ is $m/2$ since $\kappa \in [0, m]$. We need to average P over many samples and define the estimate of the storage capacity for a system of size N_s

$$\alpha(N_s) = \frac{\langle P \rangle}{N_s}. \tag{B.51}$$

For the numerical simulations in Fig. D.1, N_s is equal to N and m for the classical and quantum percepton, respectively. Moreover, we have 2^N and 2^m realizations of the weights for the classical and quantum models, respectively. We used 10000 samples for each simulation and we performed them three times to estimate the error. In Fig. D.1, we choose odd values of N to always have ±1 for the output and we use $m = 8, 16, 32$ since for larger m the computation becomes intractable.

B.5 Speculations about the Gaussian inputs

In the derivation of the basic expression

$$g = \alpha \left[-\frac{\kappa}{2} - \frac{1}{2} q^2 (2+\kappa)^2 \right] + \frac{1}{2} \log (1-q) + \frac{1}{2\,(1-q)}, \tag{B.52}$$

where we used an expansion in q, eliminating a part of the singular behavior at $q \to 1$. The next order contribution in the effective potential is presumably

$$\left[-\frac{\kappa}{2} - \frac{1}{2} q^2 (2+\kappa)^2 \right] \approx \left[-\frac{\kappa}{2} + \frac{1}{2}(2+\kappa)^2 - \frac{1}{2(1-q^2)}(2+\kappa)^2 \right]$$

and suggests that

$$\alpha_c (2+\kappa)^2/2 = 1,$$

implying maximal $\alpha_c(\kappa = 0) = 1/2$.

Appendix C

C.1 Appendix

C.1.1 Parameters

The parameters of the four arbitrary single qubit gates $[\phi_i, \theta_i, \omega_i]$ of Teacher RU are

$$[[2.71, 6.15, 0.42], [3.26, 1.13, 6.10], \\ [0.71, 2.54, 4.63], [4.43, 2.66, 2.18]],$$

while for the Student RU we have

$$[[-1.30, 0.40, -1.86], [0.60, 1.89, 0.37], \\ [0.17, 2.69, 0.67], [0.49, -0.14, 0.68]].$$

This is one of the many local minima that the training of Student RU could reach.

C.1.2 Additional plots

In Fig. C.1, we show the prediction maps and loss curves for the toy model with the QP (Fig. 3.1a) as the teacher and the students (Fig. 3.1a and b).

In Fig. C.2 and C.3 we present the prediction maps and loss curves with binary labels of the teachers for the students of the toy model (Fig. 3.1) and the deep-shallow architectures (Fig. 3.3), respectively. In Fig. C.4, we present the alteration of the QP with many processing and entangling gates. This alteration does not improve the performance of the model. This again validates that the functions that can be learned are definitely determined by the number of times the data are encoded in the circuit **?** and not the amount of deferred measurements.

C.1.3 Labelling matters

In Fig. C.5, we show how the labelling affects the results. We plot the prediction map for the circular data with labels -1 for the inner (black) and 1 for the outer circles (yellow) (Student 1 in Fig. C.5). Then, we plot the prediction map for the student with opposite labelling of the data, i.e. 1 for the inner (black) and -1 (yellow) for the outer circles (Student 2 in Fig. C.5). As shown, the prediction map of Student 2 with RX is very different from the circular structure of the data and the training is not successful.

The final state of the ancilla qubit is of the form $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ and the measurement gives $\langle Z\rangle = \alpha^2 - \beta^2$. Therefore, for the label -1, $\langle Z\rangle$ should be close to -1. The average probability vector of the ancilla qubit for the label -1 is $P_{-1} = [\alpha^2, \beta^2] = [0.37, 0.64]$, and indeed, $\alpha^2 - \beta^2 < 0$. Then, the training is successful and the prediction map of Student 1 is similar to the data structure. If we change the labelling, we get the following average probability vector $P_{-1} = [0.94, 0.06]$ for the label -1, where now $\alpha^2 - \beta^2 > 0$. This is the reason why the training fails for Student 2, which is fixed by adding a PauliX gate before the measurement (see Student - Re-uploading with RX in Fig. C.5).

C.1.4 Input data

As emphasized in the main text, the normalization of the input data highly affect the training and success of the circuits, since the quantum functions are periodic in the input data. For example, we train the students in Fig. 3.3, but with a different normalization, i.e. instead of $[-\pi, \pi]$ with $[-1, 1]$. As it can be seen in Fig. C.6, the prediction maps do not show the whole structure of the data and both students seem to perform well, contrary to Fig. 3.3 where Student - Re-uploading performs better. Therefore, one should be careful when choosing the normalization of the input data.

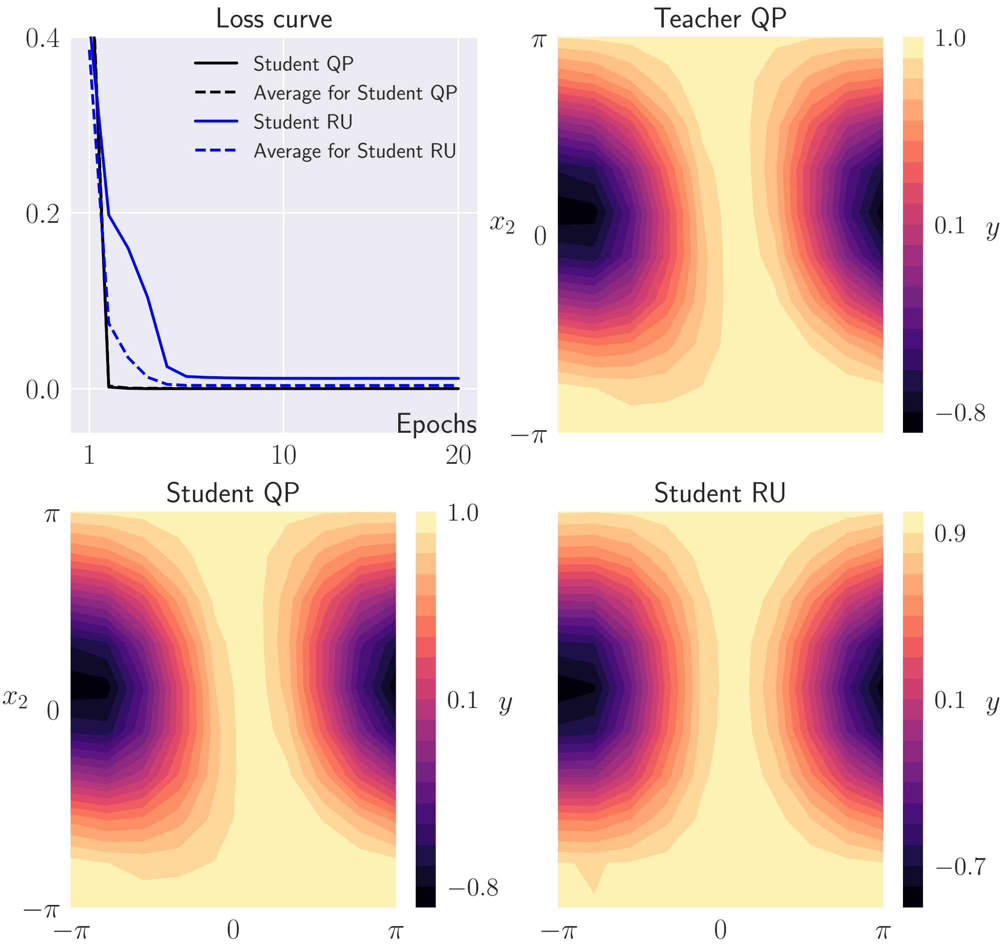


Figure C.1 Teacher-student training for teacher (Fig. 3.1a) and the two students (Fig. 3.1 a and b). The prediction maps show one particular example of the training with the corresponding losses (solid loss curves). The average loss (dashed lines) show the average over all 10 random initializations.

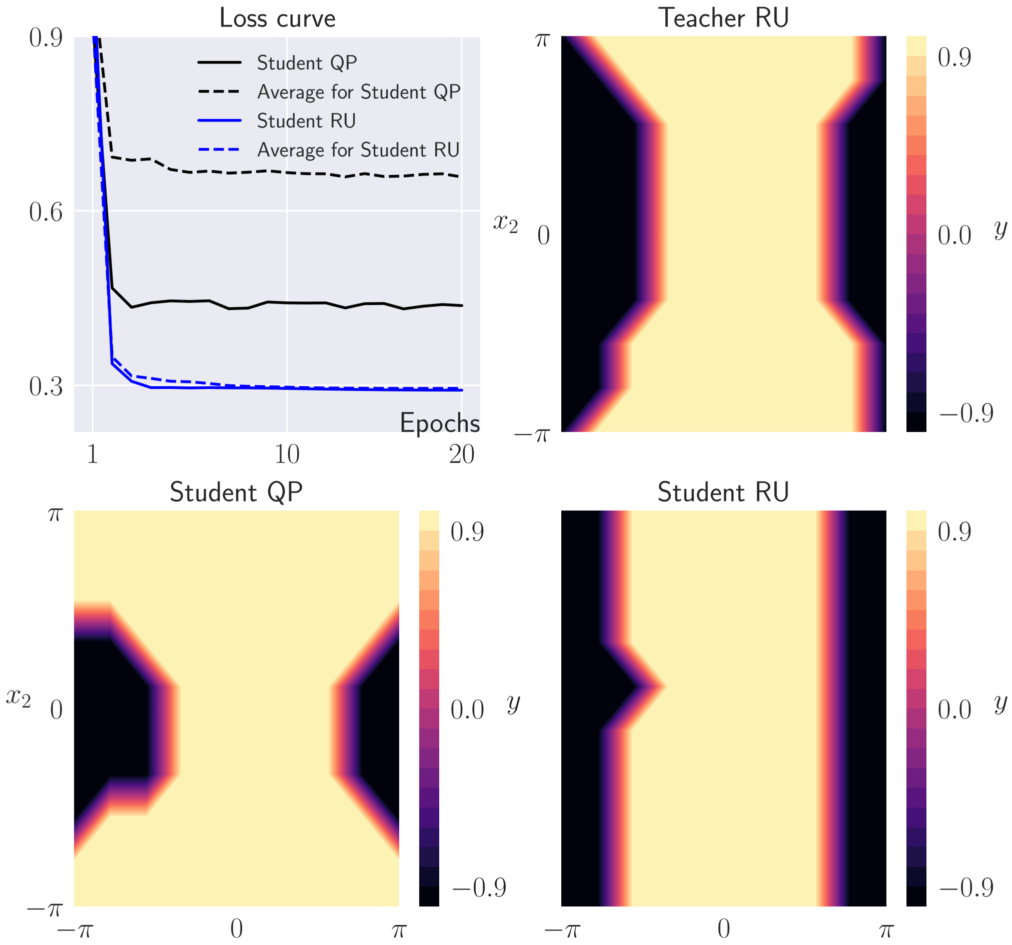


Figure C.2 The binary prediction maps of the teacher RU and the students: Student - QP and Student - RU as in Fig. 3.1. In the upper right corner, we plot the accuracy curves for the QP (Student - QP black line), RU (Student - Re-uploading blue line) and the student averages over 10 different realizations of the teacher (Student - QP black dashed line and Student - Re-uploading blue dashed line).

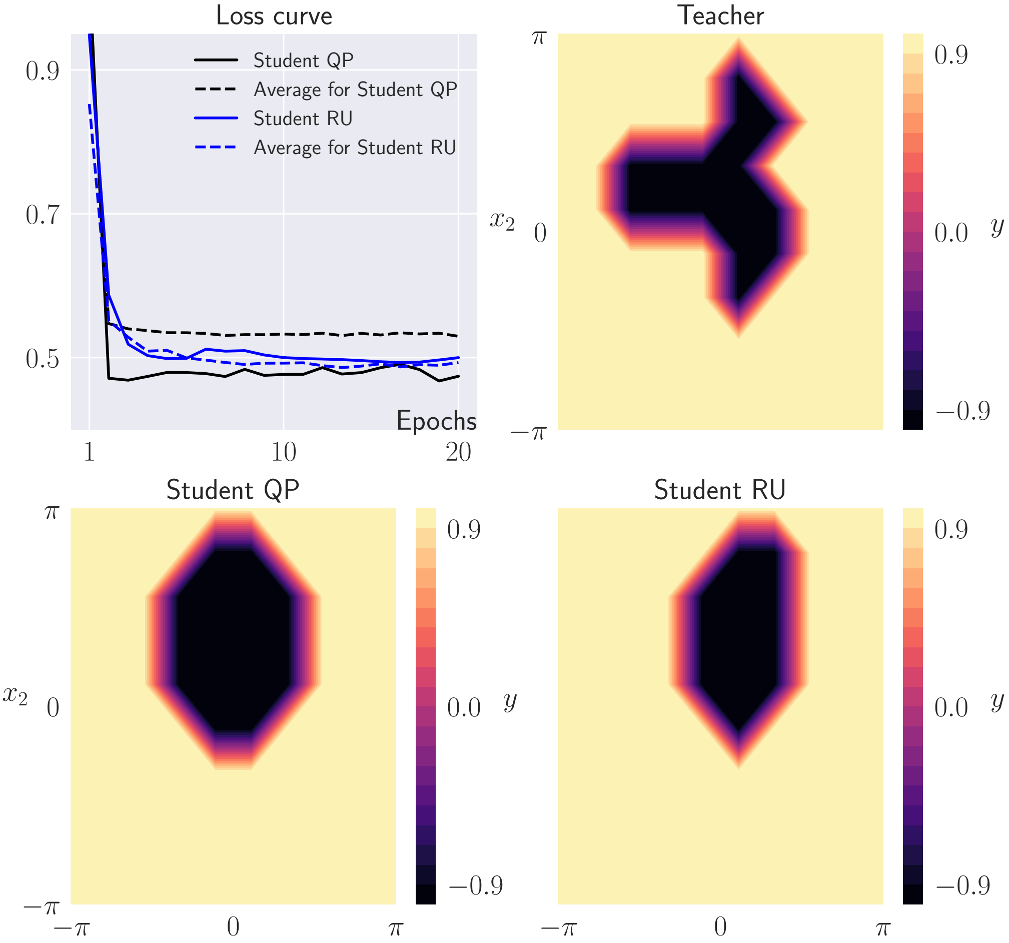


Figure C.3 The binary prediction maps of the teacher with the deep architecture in Fig. 3.3 and the students: Student - QP and Student - RU as in Fig. 3.1. In the upper right corner, we plot the accuracy curves for the QP (Student - QP black line), RU (Student - Re-uploading blue line) and the student averages over 10 different realizations of the teacher (Student - QP black dashed line and Student - Re-uploading blue dashed line).

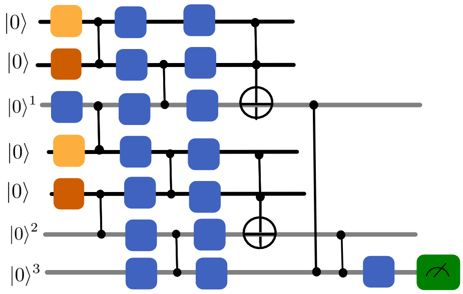


Figure C.4 An alteration of the QP with many processing and entangling gates.

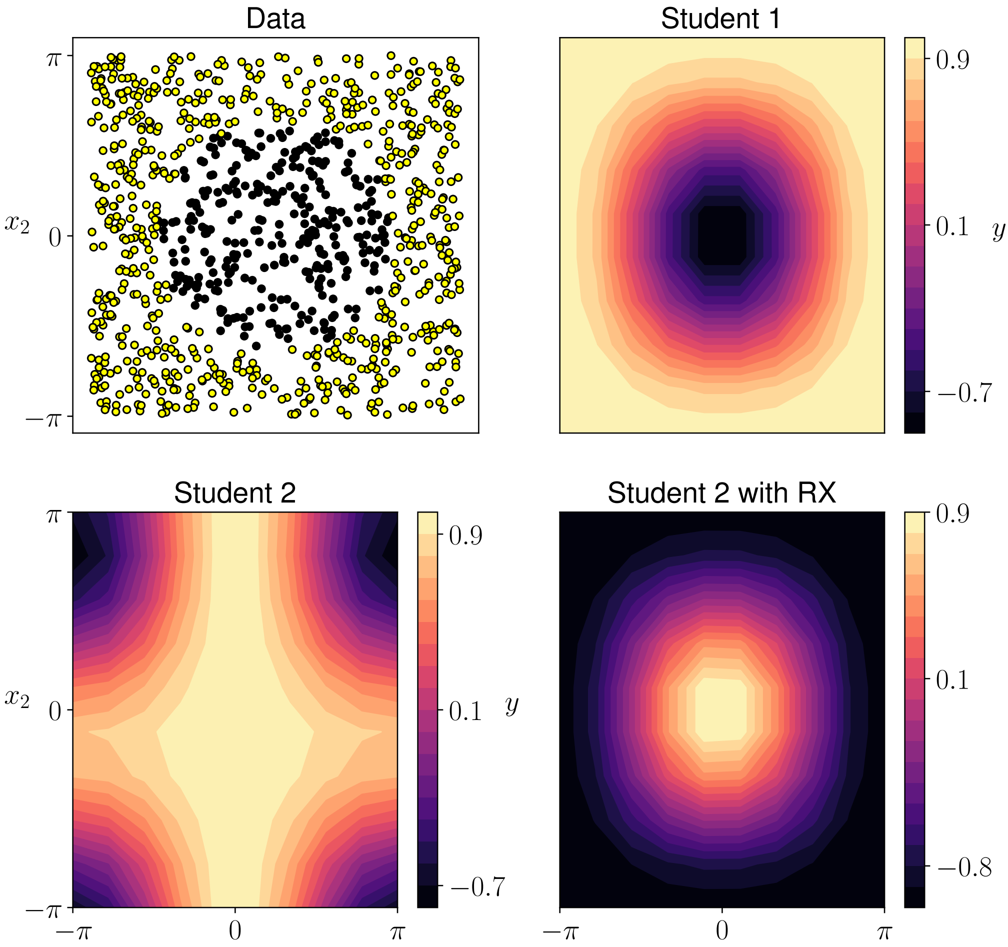


Figure C.5 We present the input data along with the prediction maps for QP students with different labelling, i.e. Student 1 has labelling -1 for the inner (black) and 1 for the outer circles (yellow), while Student 2 has exactly the opposite. Student 2 with RX has the same labelling as Student 2, but a RX gate is applied before the measurement.

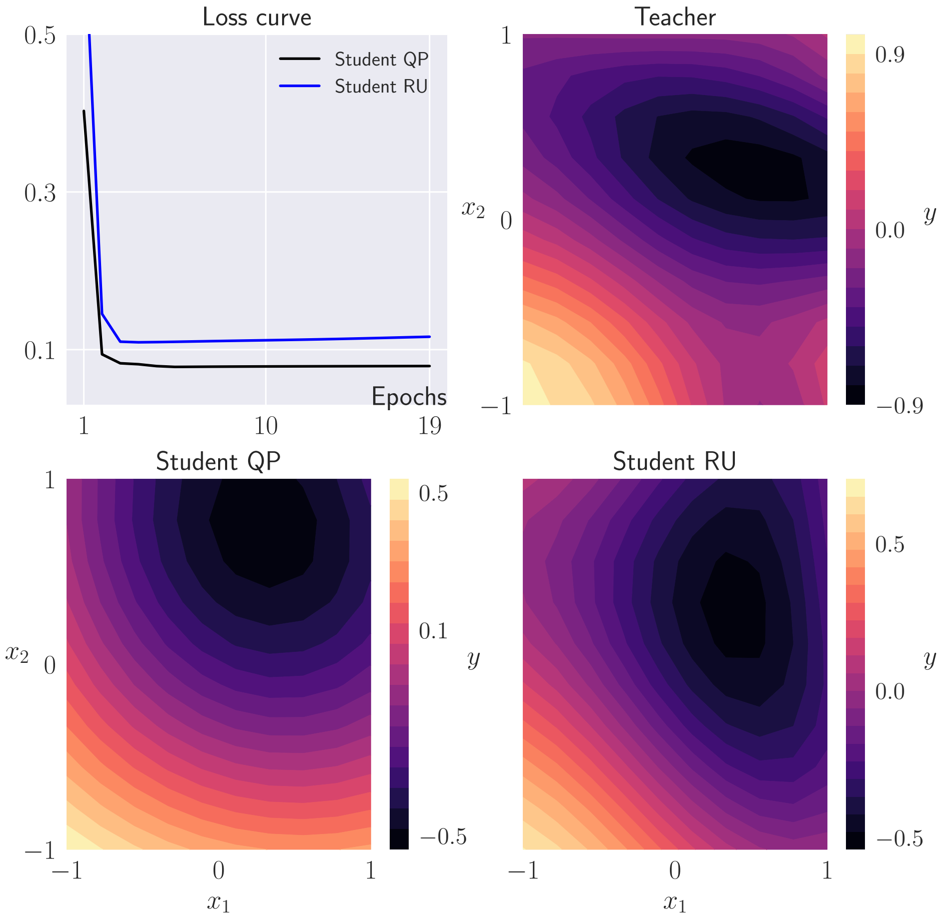


Figure C.6 The prediction maps of the teacher with the deep architecture in Fig. 3.3 and the students: Student - QP and Student - RU as in Fig. 3.1 but with normalization $[-1, 1]$ instead of $[-\pi, \pi]$. In the upper right corner, we plot the loss curves for the QP (Student - QP black line) and the RU (Student - Re-uploading blue line).

Appendix D

D.1 Monte-Carlo integration

For the numerical simulation we use Monte-Carlo integration McClean and approximate the integral as

$$\int_{U(N)} f(U)dU \approx \frac{1}{p}\sum_{i=1}^{p} f\left(U_i\right), \tag{D.1}$$

where p is the total number of random unitaries used and U_i is a randomly drawn unitary according to the Haar measure. To define such unitary numerically, we start by calling a $N \times N$ matrix with Gaussian values. Then, we perform a QR decomposition on this matrix which gives two matrices Q and R. Next, we define the diagonal matrix D from the diagonal elements of the matrix R, i.e. $D_{ii} = R_{ii}/|R_{ii}|$. Finally, the unitary random matrix according to the Haar measure is defined as $U_i = QD$ McClean.

D.2 Prediction maps

Fig. D.1 shows an example of the prediction maps for the 2-qubit circuit as a teacher and the 3-qubit circuit as a student along with the reverse roles for each circuit. In both cases, the student deviates from the data distribution of its teacher.

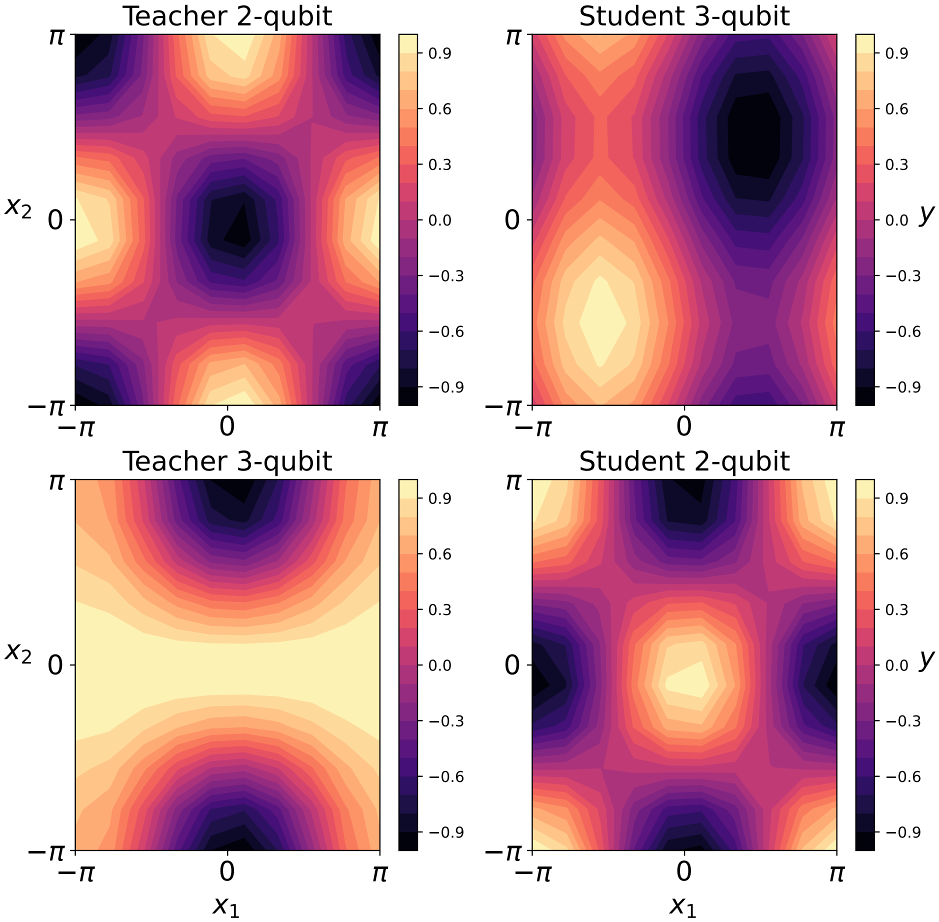


Figure D.1 Examples of the prediction maps for the 2-qubit and 3-qubit circuits at the roles of the teacher and student. In both cases, the students have significant discrepancies with their teachers.

D.3 Coefficient formula

Here, we present the explicit formula of the real and imaginary parts of the coefficient c_{11} for the 2-qubit and 3-qubit circuits in Fig. 4.2 in Eq. (D.2),(D.3) and Eq. (D.4),(D.5), respectively for the real and imaginary parts.

$$Re\left(c_{11}\right) = \sin\left(\frac{\theta_2}{2}\right)\sin\left(\frac{\phi_1}{2}+\frac{\omega_1}{2}\right)\cos\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_2}{2}\right)\sin\left(\frac{\theta_1}{2}\right)\cos\left(\frac{-\phi_1}{2}+\frac{\omega_1}{2}\right)$$
$$\left(\sin\left(\frac{\phi_2}{2}+\frac{\omega_2}{2}\right)\cos\left(-\frac{\phi_2}{2}+\frac{\omega_2}{2}\right)-\cos\left(\frac{\phi_2}{2}+\frac{\omega_2}{2}\right)\sin\left(-\frac{\phi_2}{2}+\frac{\omega_2}{2}\right)\right)$$
$$-\sin\left(\frac{\theta_2}{2}\right)\sin\left(-\frac{\phi_1}{2}+\frac{\omega_1}{2}\right)\cos\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_2}{2}\right)\sin\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\phi_1}{2}+\frac{\omega_1}{2}\right)$$
$$\left(\sin\left(\frac{\phi_2}{2}+\frac{\omega_2}{2}\right)\cos\left(-\frac{\phi_2}{2}+\frac{\omega_2}{2}\right)-\cos\left(\frac{\phi_2}{2}+\frac{\omega_2}{2}\right)\sin\left(-\frac{\phi_2}{2}+\frac{\omega_2}{2}\right)\right)+$$
$$\frac{1}{4}\left(-4\cos\left(\frac{\theta_2}{2}\right)^2+2\right)\cos\left(\frac{\theta_1}{2}\right)^2+\frac{1}{2}\cos\left(\frac{\theta_2}{2}\right)^2-\frac{1}{4} \tag{D.2}$$

$$Im\left(c_{11}\right) = -\left(\cos\left(\frac{\theta_2}{2}\right)^2-\frac{1}{2}\right)\sin\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\phi_1}{2}+\frac{\omega 1}{2}\right)\cos\left(\frac{\theta_1}{2}\right)\cos\left(-\frac{\phi_1}{2}+\frac{\omega 1}{2}\right)+\Bigg(\mathrm{c}$$

$$Re\left(c_{11}\right) = -\frac{1}{8}-\frac{1}{8}\left(-2+4\cos\left(\frac{\theta_2}{2}\right)^2\right)\cos\left(\frac{\theta_1}{2}\right)^2+$$
$$\frac{1}{2}\sin(\phi_2)\sin(\theta_1)\sin\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right)\cos\left(\frac{\theta_2}{2}\right)\cos\left(\frac{\theta_1}{2}\right)+\frac{1}{4}\cos\left(\frac{\theta_2}{2}\right)^2 \tag{D.4}$$

$$Im\left(c_{11}\right) = \frac{1}{2}\sin(\phi_2)\sin\left(\frac{\theta_2}{2}\right)\cos\left(\frac{\theta_2}{2}\right)\cos\left(\frac{\theta_1}{2}\right)^2-$$
$$\frac{1}{2}\sin(\phi_1)\left(\cos\left(\frac{\theta_2}{2}\right)^2-\frac{1}{2}\right)\sin\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_1}{2}\right)+\frac{1}{4}\sin\left(\frac{\theta_2}{2}\right)\cos\left(\frac{\theta_2}{2}\right)\sin(\theta_2) \tag{D.5}$$

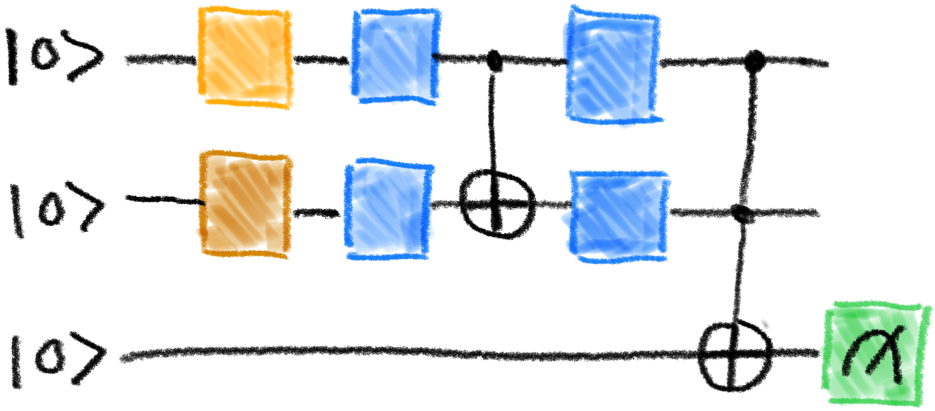


Figure D.2 A 3-qubit circuit with an extra layer L marked by the dashed lines.

D.4 3-qubit circuit with 4 gates

Here, we present the Fourier coefficients in Fig. D.3 for the circuit with 3 qubits and four parameterized gates sketched in Fig. D.2. We see that the coefficients $c_{1,1}$ and $c_{1,-1}$ now take similar values with their corresponding coefficients shown in Fig. 4.5 in orange color.

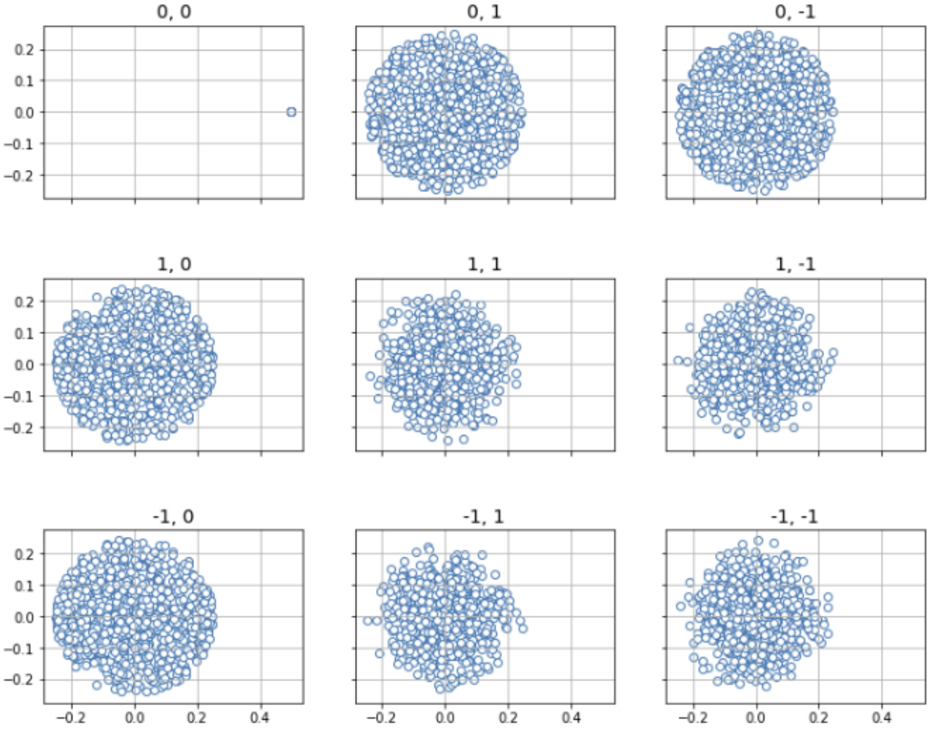


Figure D.3 The real and imaginary part of the Fourier coefficients for the 3-qubit (light blue color) circuit of Fig. D.2.

Appendix E

E.1 STO-3G

Here, we explore the behaviour of the H_n with the STO-3G basis set in terms of the ansatz overlap as a function of the system size $n \in \{2, 4, 6, 8\}$. In Fig. E.1, we see that HF fidelities drop faster than SPA, which on the contrary gives reasonable high fidelity values for all studied n.

E.2 Booster resource estimation analysis

Following the analysis of the booster algorithm Wang *et al.* [2022b], the booster operation $f(H)$ of the given Hamiltonian H could be implemented by obtaining a Fourier approximation of f and implementing a linear combination of unitaries (LCU) method as

$$f_D(x) = \int_{-D}^{D} \hat{f}(\xi) e^{i2\pi x\xi} d\xi. \tag{E.1}$$

According to the aforementioned work, the proxy depth of this operation could be estimated as $2D$. Next, the integral could be further discretized as

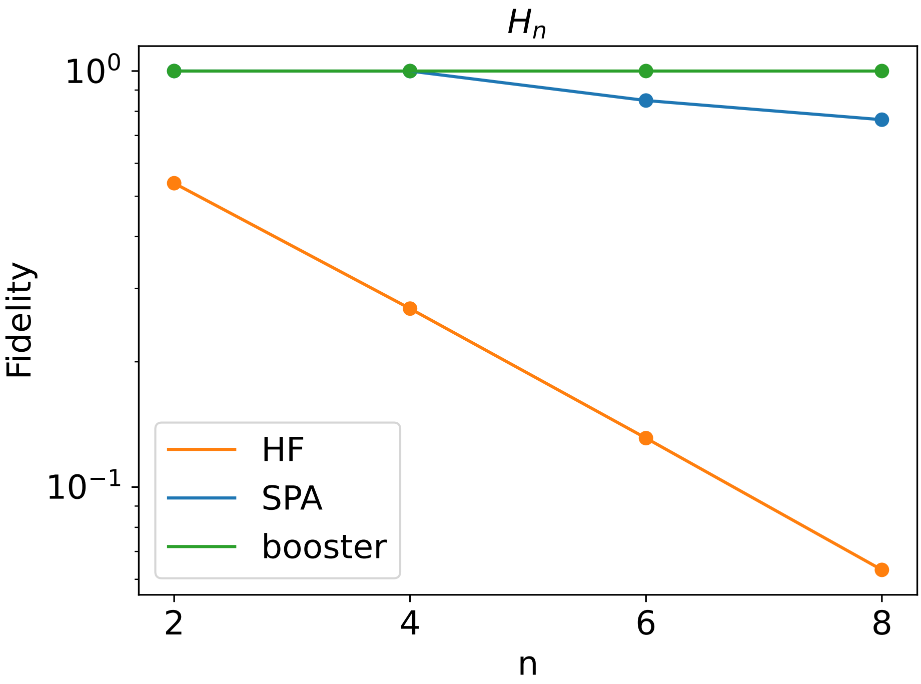


Figure E.1 Here we plot the HF, SPA and booster overlap squared for the H_n systems with STO-3G basis set with an increasing system size n.

$$f_{D,N}(x) = \frac{D}{N} \sum_{j=-N}^{N-1} \hat{f}\left(\xi_j\right) e^{i2\pi x \xi_j}, \tag{E.2}$$

with $\xi_j = (j+1/2)\frac{D}{N}$ for $j = -N, -N+1, \ldots, N-1$. Next, assuming a Gaussian booster $exp\left(-ax^2\right)$ the above equation becomes

$$f_{D,N;a}(x) = \frac{D}{N}\sqrt{\frac{\pi}{a}} \sum_{j=-N}^{N-1} e^{-\frac{\left(\pi\xi_j\right)^2}{a}} e^{2\pi i x \xi_j}, \tag{E.3}$$

where $\xi_j = (j+1/2)D/N$, and N is sufficiently large so that $f_a(x) \approx f_{D;a}(x) \approx f_{D,N;a}(x)$. Finally, we can apply a Trotter decomposition to the operation $e^{2\pi i H \xi_j}$ of the given Hamiltonian H for one Trotter step to estimate the Pauli rotations, and in turn, the T-gate count $T_{K=1}$. Usually, a required number of Trotter steps K is necessary to reach the desired precision. Thus, the total number of T-gates for the booster algorithm is given by $T = 2DKT_{K=1}$.

Finally, following the work of Gratsea *et al.* [2022], the booster algorithm requires a number of repetitions $1/P_{succ}$ to ensure their success, where the success probability P_{succ} is given by

$$p_{succ}\left(f_{T,N}\right) \approx \frac{\left\langle \psi \left| f^2(H) \right| \psi \right\rangle}{f^2(0)} = \left\langle \psi \left| f^2(H) \right| \psi \right\rangle, \tag{E.4}$$

since $f^2(0) = 1$ for the Gaussian booster.

E.3 Number of Trotter steps

The work of von Burg *et al.* [2021] estimates the necessary number of Trotter steps for 1-D Hydrogen chains to chemical accuracy. To apply this analysis to the operation $e^{2\pi i H \xi_j}$ discussed in the previous section and Eq. (E.3), we need that the maximal simulation time of the aforementioned operation is smaller than the time $\epsilon^{-1} = 625$ considered in the work von Burg *et al.* [2021], where ϵ is the chemical accuracy $1.6mHa$.

We estimate the maximal simulation time by summing over the absolute value of $x_j = (j + 1/2)D/N$ from $j = -N$ to $N - 1$, which results to $\frac{\pi D}{\Delta}$. The proxy depth D is a hyper-parameter in the optimization of the booster and is set to 10. The spectral gap Δ is equal to $[1.1580, 2.7287, 4.48, 6.35]$ for the hydrogen chains with $n = [2, 4, 6, 8]$, which results to the maximal simulation time $[27, 12, 7, 2]$, respectively. This suggests that the maximal simulation time is smaller than the simulation time considered in the work of von Burg *et al.* [2021] and the results presented there are applicable in this analysis.

The authors present an empirical estimation of the number of Trotter steps necessary for the linear hydrogen chains with bond distance $d = 1.7$ and find that approximately $K = 10$ number of Trotter steps are required. According to the work of Tranter *et al.* [2019], increasing the bond distance decreases the Trotter error, and therefore, in our resource estimations of linear hydrogen chains with $d = 3.0$ the empirical calculation of $K = 10$ Trotter steps is a good proxy (see Fig. 5.6, Table 5.2).

Finally, the authors of the work von Burg *et al.* [2021] observe a $10\sqrt{n}$ reduction in the number of Trotter steps required compared to a rigorous bound analysis given in Cohen *et al.* [2012]. In Fig. E.2, we present the runtime ratios while taking into account the increased number of Trotter steps $K = 100\sqrt(n)$ for the booster algorithm instead of $K = 10$ in Fig. 5.6.

E.4 Overlap values

In Table E.1 we presented the overlap squared values used in Fig. 5.5.

Table E.1 We present overlap squared values used in Fig. 5.5 for H_n.

H_n	γ_0^2	γ_{SPA+X}^2	γ_b^2
H_2	0.63	0.99	1.0
H_4	0.0375	0.69	1.0
H_6	0.0052	0.94	1.0
H_8	0.00073	0.19	1.0

E.5 Failure tolerance

Usually δ is chosen through the maximal probability with which ones allow the algorithm to fail. Therefore, we define the failure

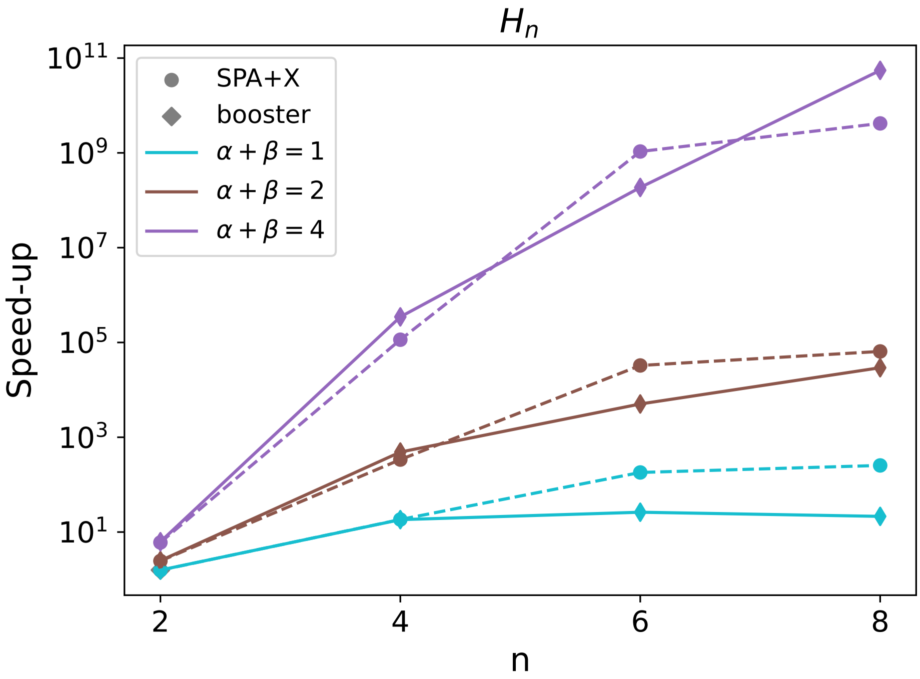


Figure E.2 Here we plot the ratios T_o/T for SPA and booster over HF with $K = 100\sqrt{n}$ of H_n system for an increasing system size $n = [2, 4, 6, 8]$ and bond distance $d = 3.0$.

tolerance δ_C as

$$\delta_C = 1 - (1-\delta)^R \approx R\delta \Rightarrow \delta \approx \delta_C \times (R)^{-1}\,, \tag{E.5}$$

where R is the number of Pauli Rotations and δ is the necessary precision for operating the R number of gates.

E.6 Computational details

We compiled the SPA-variant circuits of Fig. 5.5 to Pauli Rotations R_{SPA} and CNOT gates by using the circuit compilation incorporated in TEQUILA Kottmann *et al.* [2021a]. In Table E.3, we present the number of Pauli Rotations R_{SPA} and the respective values of δ_{SPA} for the T-gate counts of the SPA-variant circuits also presented in Table 5.1.

Table E.2 We present the Pauli rotations (R_{SPA}, R_{SPA+GS}), the necessary precision for operating the aforementioned number of gates (δ_{SPA}, δ_{SPA+GS}) of the SPA-variants circuits for H_n.

H_n	R_{SPA}	δ_{SPA}	R_{SPA+GS}	δ_{SPA+GS}
H_2	1	10^{-3}	21	5×10^{-5}
H_4	2	5×10^{-4}	2.7×10^{3}	8.3×10^{-6}
H_6	3	3×10^{-4}	4.2×10^{3}	3.3×10^{-6}
H_8	560	2.5×10^{-4}	2.2×10^{4}	1.8×10^{-6}

For estimating the number of T-gate counts of the booster algorithm presented in Table 5.1, we used a Trotter decomposition with $K = 1$ number of steps for the Hamiltonians of H_n by using the time evolution function in ORQUESTRA Zapata AI [2023]. Then, we exported the circuit with Qiskit Qiskit contributors [2023] and computed the number of Pauli Rotations R_b with their respective values of δ_b (presented in the Table E.3 above).

Table E.3 We present the Pauli rotations (R_b), the necessary precision for operating the aforementioned number of gates (δ_b) and T-gate counts of the booster algorithm (T_B) for H_n.

H_n	R_b	δ_b	T_B
H_2	9.3×10^2	3.3×10^{-5}	1.9×10^5
H_4	1.3×10^7	7.6×10^{-7}	1.3×10^7
H_6	7.4×10^3	1.4×10^{-7}	5.2×10^7
H_8	2.5×10^5	4.0×10^{-8}	1.6×10^8

Classical FCI and HF energies are computed with PYSCF Sun *et al.* [2018]. MRA-PNOs are computed with MADNESS Harrison *et al.* [2016] using the implementation described in Kottmann *et al.* [2021b, 2020] on top of the framework described in Harrison *et al.* [2004]; Bischoff [2014]. Exact diagonalization of Hamiltonians was performed with sparse solvers implemented in SCIPY Virtanen *et al.* [2020]. The quantum simulation backend was QULACS Suzuki *et al.* [2021].

Appendix F

F.1 Scalability of today's devices

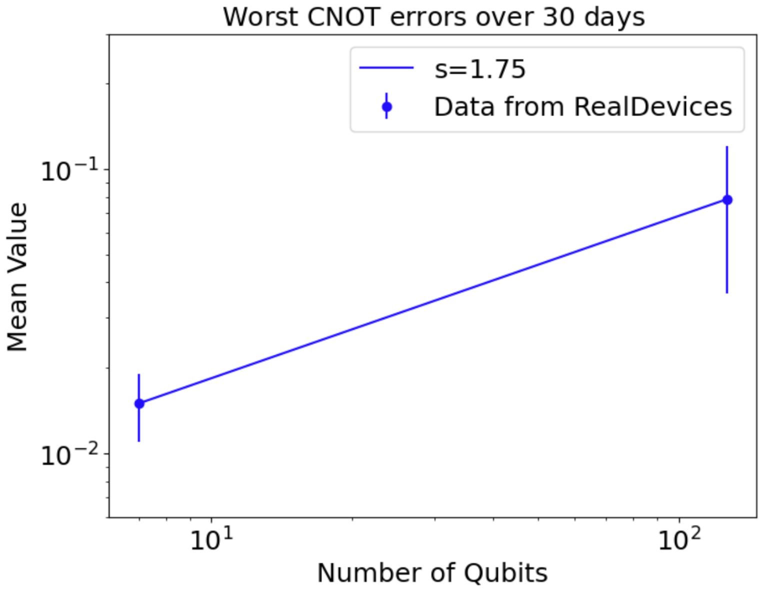


Figure F.1 We plot the worst two qubit gate error of two IBM quantum devices on the cloud as a function of the number of qubits. The power law fit (blue line) suggests today's scalability is $s = 1.75$ and $p_0 = 0.005$.

Here we estimate the scalability of today's quantum devices. To this end, we collected data from two IBM devices available on the cloud, namely *lagos* and *brisbane*. We collected the CNOT error rates at 10:00 am each day over 30 days. In Fig. F.1, we plot the mean and standard deviation of the worst-case CNOT error rates for the studied devices. We then make a power-law fit of the data to estimate the p_0 and s as introduced in Eq. 6.3. We find that $p_0 = 0.005$ and $s = 1.75$, which we refer to as today's scalability. Figure 6.1 shows this point to lie in the NISQ regime, despite p_0 being below threshold.

F.2 Logarithmic scalability model

In this section we investigate the implications of a more optimistic scalability model. Instead of the power law model of Equation 6.3, we consider a logarithmic model for the scalability profile,

$$p_{\text{phys}}(Q_{\text{phys}}; \mathcal{V}) = p_0 \left(1 + \frac{1}{\sigma} \ln(Q_{\text{phys}})\right), \tag{F.1}$$

where σ is the scalability parameter analogous to s in Equation 6.3. The parameterization is chosen such that at $Q_{\text{phys}} = 1$, the function value and slope match those of Equation 6.3. With this alternative scalability model, the physical qubit number at which the physical error rate exceeds the threshold value is now

$$Q_{\text{phys}}^{\max} = \exp\left(\sigma \frac{p_0 - p_{\text{th}}}{p_{\text{th}}}\right), \tag{F.2}$$

which, compared to the power law model, grows exponentially in the gap between p_0 and p_{th}.

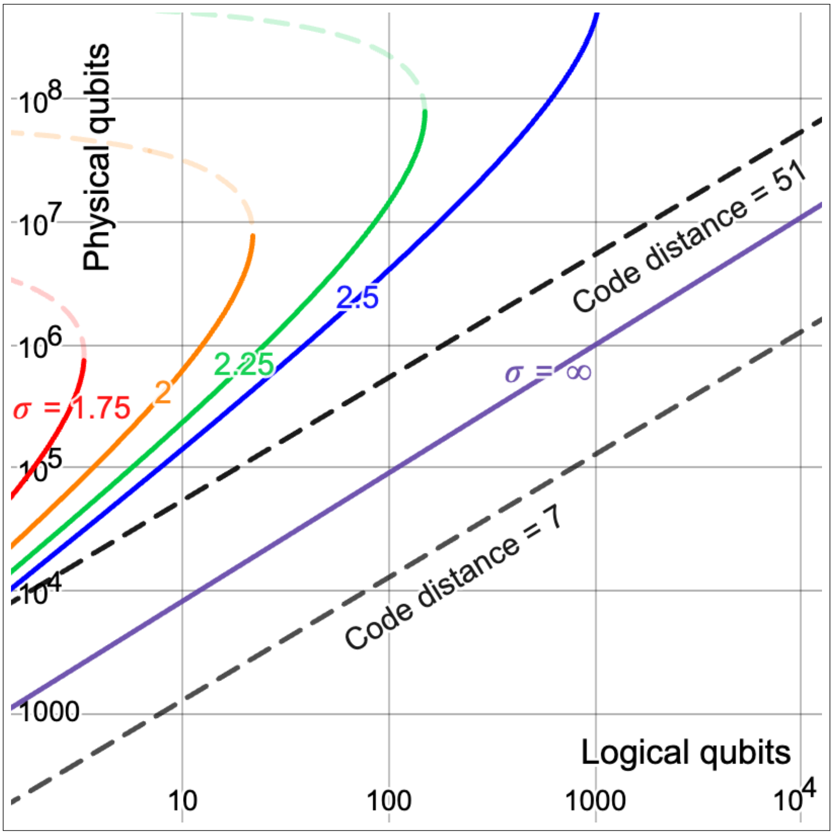


Figure F.2 This plot shows the QPE resource overhead under the logarithmic scalability model of Eq. F.1. Similar to the power law model shown in Figure 6.4, the logarithmic scalability model also predicts that, for each finite value of scalability parameter σ, there is a maximum problem instance size that can be accommodated by the architecture. However, the logarithmic scalability model is more optimistic in that, for the same base error rate $p_{\text{phys}}(Q_{\text{phys}} = 1) = p_0$ and for the same (logarithmic) slope at $Q_{\text{phys}} = 1$, the maximum problem instance size is far larger for the logarithmic scalability model. Thus, in order to observe the limited problem instance size in the range of 10 to 10,000 logical qubits, we use the larger base error rate of $p_0 = 0.001$ compared to the $p_0 = 0.0001$ used in Figure 6.4. An editable version of the plot can be accessed here: https://www.desmos.com/calculator/cnh0vchq6l

Bibliography

(Accessed 2023). Quantum algorithms zoo. https://quantumalgorithmzoo.org/.

Aaronson, S. and Arkhipov, A. (2013). The computational complexity of linear optics. *Theory of Computing*, **9**, 143–252.

Abbas, A., Sutter, D., Zoufal, C., Lucchi, A., Figalli, A. and Woerner, S. (2021). The power of quantum neural networks. *Nature Computational Science*, **1**, 403–409.

Aguilar, L., Barrera, D., García, R., Grana, M., Mora, M., Rocha, M. and Tavares, J., eds. (2020). *Computer Science - CACIC 2019: 25th Argentine Congress, Córdoba, Argentina, October 28–November 1, 2019, Proceedings*, vol. 1089 of *Communications in Computer and Information Science*. Springer.

Aharonov, D. and Ben-Or, M. (2008a). Fault-tolerant quantum computation with constant error rate. *https://doi.org/10.1137/S0097539799359385*, **38**, 1207–1282.

Aharonov, D. and Ben-Or, M. (2008b). Fault-tolerant quantum computation with constant error rate. *SIAM Journal on Computing*, **38**, 1207–1282.

Aharonov, D., Kitaev, A. and Preskill, J. (2006). Fault-tolerant quantum computation with long-range correlated noise. *Physical Review Letters*, **96**, 050504.

AI, G.Q. (2022). Suppressing quantum errors by scaling a surface code logical qubit.

Akahoshi, Y., Maruyama, K., Oshima, H., Sato, S. and Fujii, K. (2023). Partially fault-tolerant quantum computing architecture with error-corrected clifford gates and space-time efficient analog rotations. *arXiv preprint arXiv:2303.13181*.

Albash, T. and Lidar, D.A. (2018). Adiabatic quantum computation. *Reviews of Modern Physics*, **90**.

Alcazar, J. and Perdomo-Ortiz, A. (2021). Enhancing combinatorial optimization with quantum generative models.

Alcazar, J., Cadarso, A., Katabarwa, A., Mauri, M., Peropadre, B., Wang, G. and Cao, Y. (2022). Quantum algorithm for credit valuation adjustments. *New Journal of Physics*, **24**, 023036.

Aliferis, P. and Leung, D.W. (2006). Simple proof of fault tolerance in the graph-state model. *Physical Review A*, **73**, 032308.

Aliferis, P. and Preskill, J. (2008). Fault-tolerant quantum computation against biased noise. *Physical Review A - Atomic, Molecular, and Optical Physics*, **78**, 052331.

Aliferis, P., Gottesman, D. and Preskill, J. (2006). Quantum accuracy threshold for concatenated distance-3 code. *Quantum Information and Computation*, **6**, 97–165.

Aliferis, P., Gottesman, D. and Preskill, J. (2008). Accuracy threshold for postselected quantum computation. *Quantum Information & Computation*, **8**, 181–244.

Altaisky, M.V. (2001). Quantum neural network (preprint). *arXiv preprint quant-ph/0107012*.

Amit, D. (1989). *Modelling Brain Function*. Cambridge University Press.

Amsler, M., Deglmann, P., Degroote, M., Kaicher, M.P., Kiser, M., Kühn, M., Kumar, C., Maier, A., Samsonidze, G., Schroeder, A., Streif, M., Vodola, D. and Wever, C. (2023). Quantum-enhanced quantum monte carlo: an industrial view.

Andrecut, M. and Ali, M.K. (2003). Quantum Associative Memory. *International Journal of Modern Physics B*, **17**, 2447–2472.

Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J.C., Barends, R., Biswas, R., Boixo, S., Brandao, F.G., Buell, D.A. *et al.* (2019a). Quantum supremacy using a programmable superconducting processor. *Nature*, **574**, 505–510.

Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J.C., Barends, R., Biswas, R., Boixo, S., Brandao, F.G.S.L., Buell, D.A. and et al. (2019b). Quantum supremacy using a programmable superconducting processor. *Nature*, **574**, 505–510.

Aspuru-Guzik, A., Dutoi, A.D., Love, P.J. and Head-Gordon, M. (2005). Simulated quantum computation of molecular energies. *Science*, **309**, 1704–1707.

Aubin, B., Perkins, W. and Zdeborová, L. (2019). Storage capacity in symmetric binary perceptrons. *Journal of Physics A: Mathematical and Theoretical*, **52**, 294003.

Auffè ves, A. (2022). Quantum technologies need a quantum energy initiative. *PRX Quantum*, **3**.

Ba, J. and Caruana, R. (2014). Do deep nets really need to be deep? In *Advances in neural information processing systems (NIPS)*.

Babbush, R., Love, P. and Aspuru-Guzik, A. (2014). Adiabatic quantum simulation of quantum chemistry. *Scientific Reports*, **4**, 6603.

Babbush, R., Gidney, C., Berry, D.W., Wiebe, N., McClean, J., Paler, A., Fowler, A. and Neven, H. (2018a). Encoding electronic spectra in quantum circuits with linear t complexity. *Physical Review X*, **8**.

Babbush, R., Gidney, C., Berry, D.W., Wiebe, N., McClean, J., Paler, A., Fowler, A. and Neven, H. (2018b). Encoding electronic spectra in quantum circuits with linear t complexity. *Physical Review X*, **8**.

Babbush, R., Huggins, W.J., Berry, D.W., Ung, S.F., Zhao, A., Reichman, D.R., Neven, H., Baczewski, A.D. and Lee, J. (2023). Quantum simulation of exact electron dynamics can be more efficient than classical mean-field methods.

Bao, S., Kleer, S., Wang, R. and Rahmani, A. (2018). Optimal control of superconducting gmon qubits using pontryagin's minimum principle: Preparing a maximally entangled state with singular bang-bang protocols. *Physical Review A*, **97**, 062343.

Beer, K., Bondarenko, D., Farrelly, T., Osborne, T.J., Salzmann, R., Scheiermann, D. and Wolf, R. (2020). Training deep quantum neural networks. *Nature Communications*, **11**.

Begušić, T. and Chan, G.K. (2023). Fast classical simulation of evidence for the utility of quantum computing before fault tolerance. *arXiv preprint arXiv:2306.16372*.

Benatti, F., Gramegna, G. and Mancini, S. (2022). Pattern capacity of a single quantum perceptron. *Journal of Physics A: Mathematical and Theoretical*.

Benedetti, M., Garcia-Pintos, D., Perdomo, O., Leyton-Ortega, V., Nam, Y. and Perdomo-Ortiz, A. (2019a). A generative modeling approach for benchmarking and training shallow quantum circuits. *npj Quantum Information*, **5**.

Benedetti, M., Garcia-Pintos, D., Perdomo, O., Realpe-Gómez, J., Biswas, R., Fagotti, M., Leichenauer, S., Lubasch, M., Keesling, A., Cincio, L., Hen, I., Sanz, M., Sutor, A., Van Meter, R., Wiebe, N., Yamamoto, Y. and Troyer, M. (2019b). A generative modeling approach for benchmarking and training shallow quantum circuits. *npj Quantum Information*, **5**, 45.

Bernstein, E. and Vazirani, U. (1993). Quantum complexity theory. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, 11–20.

Berry, D.W., Higgins, B.L., Bartlett, S.D., Mitchell, M.W., Pryde, G.J. and Wiseman, H.M. (2009). How to perform the most accurate possible phase measurements. *Phys. Rev. A*, **80**, 052114.

Berry, D.W., Childs, A.M. and Kothari, R. (2015). Hamiltonian simulation with nearly optimal dependence on all parameters. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, IEEE.

Berry, D.W., Gidney, C., Motta, M., McClean, J.R. and Babbush, R. (2019). Qubitization of arbitrary basis quantum chemistry leveraging sparsity and low rank factorization. *Quantum*, **3**, 208.

Beverland, M.E., Murali, P., Troyer, M., Svore, K.M., Hoefler, T., Kliuchnikov, V., Low, G.H., Soeken, M., Sundaram, A. and Vaschillo, A. (2022). Assessing requirements to scale to practical quantum advantage. *arXiv preprint arXiv:2211.07629*.

Bischoff, F.A. (2014). Regularizing the molecular potential in electronic structure calculations. I. SCF methods. *The Journal of Chemical Physics*, **141**, 184105.

Bittel, L. and Kliesch, M. (2021). Training variational quantum algorithms is np-hard. *Physical Review Letters*, **127**.

Blume-Kohout, R., Gamble, J.K., Nielsen, E., Rudinger, K., Mizrahi, J., Fortier, K. and Maunz, P. (2017). Demonstration of qubit operations below a rigorous fault tolerance threshold with gate set tomography. *Nature Communications*, **8**, 14485.

Brassard, G., Hoyer, P., Mosca, M. and Tapp, A. (2002). Quantum amplitude amplification and estimation. *Contemporary Mathematics*, **305**, 53–74.

Breuckmann, N.P. and Eberhardt, J.N. (2021). Quantum low-density parity-check codes. *PRX Quantum*, **2**, 040101.

Cai, Z., Babbush, R., Benjamin, S.C., Endo, S., Huggins, W.J., Li, Y., McClean, J.R. and O'Brien, T.E. (2022). Quantum error mitigation. *arXiv preprint arXiv:2210.00921*.

Calderbank, A.R., Rains, E.M., Shor, P.W. and Sloane, N.J. (1997). Quantum error correction and orthogonal geometry. *Physical Review Letters*, **78**, 405.

Campbell, E.T. (2021). Early fault-tolerant simulations of the hubbard model. *Quantum Science and Technology*, **7**, 015007.

Cao, Y., Guerreschi, G.G. and Aspuru-Guzik, A. (2017). Quantum neuron: an elementary building block for machine learning on quantum computers.

Cao, Y., Romero, J., Olson, J.P., Degroote, M., Johnson, P.D., Kieferová, M., Kivlichan, I.D., Menke, T., Peropadre, B., Sawaya, N.P. *et al.* (2019). Quantum chemistry in the age of quantum computing. *Chemical reviews*, **119**, 10856–10915.

Cao, Y., Romero, J., Olson, J., Degroote, M., Johnson, P.D., Kieferová, M., Kivlichan, I.D., Menke, T., Peropadre, B., Bauman, N.P., Cheng, S.K., Sim, S., Sung, K., McClean, J.M., Babbush, R., Aspuru-Guzik, A. and White, A.D. (2020). Quantum chemistry in the age of quantum computing. *Chemical Reviews*, **120**, 6908–6945.

Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L. and Zdeborová, L. (2019). Machine learning and the physical sciences. *Reviews of Modern Physics*, **91**, 045002.

Casas, B. and Cervera-Lierta, A. (2023). Multidimensional fourier series with quantum circuits. *Physical Review A*, **107**.

Ceroni, J., Stetina, T.F., Kieferova, M., Marrero, C.O., Arrazola, J.M. and Wiebe, N. (2022). Generating approximate ground states of molecules using quantum machine learning.

Chen, H., Wossnig, L., Severini, S., Neven, H. and Mohseni, M. (2018). Universal discriminative quantum neural networks.

Chen, Y., Neill, C., Roushan, P., Leung, N., Fang, M., Barends, R., Kelly, J., Campbell, B., Chen, Z., Chiaro, B., Dunsworth, A., Jeffrey, E., Megrant, A., Mutus, J.Y., O'Malley, P.J., Quintana, C.M., Sank, D., Vainsencher, A., Wenner, J., White, T.C., Geller, M.R., Cleland, A.N. and Martinis, J.M. (2014). Qubit architecture with high coherence and fast tunable coupling. *Physical Review Letters*, **113**, 220502.

Ciregan, D., Meier, U. and Schmidhuber, J. (2012). Multi-column deep neural networks for image classification.

Cleve, R., Ekert, A., Macchiavello, C. and Mosca, M. (1998). Quantum algorithms revisited. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, **454**, 339–354.

Cohen, A.J., Mori-Sánchez, P. and Yang, W. (2012). Challenges for density functional theory. *Chemical Reviews*, **112**, 289–320, pMID: 22191548.

Cohen, L.Z., Kim, I.H., Bartlett, S.D. and Brown, B.J. (2022). Low-overhead fault-tolerant quantum computing using long-range connectivity. *Science Advances*, **8**, 1717.

Cong, I., Choi, S. and Lukin, M.D. (2019). Quantum convolutional neural networks. *Nature Physics*, **15**, 1273–1278.

Cover, T.M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, 326–334.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, **2**, 303–314.

Czarnik, P., Arrasmith, A., Coles, P.J. and Cincio, L. (2021). Error mitigation with clifford quantum-circuit data. *Quantum*, **5**, 592.

Dallaire-Demers, P.L. and Killoran, N. (2018). Quantum generative adversarial networks. *Physical Review A*, **98**, 012324.

Davey, N. and Adams, R. (2004). High capacity associative memories and connection constraints. *Connection Science*, **16**, 47–65.

Debnath, S., Linke, N.M., Figgatt, C., Landsman, K.A., Wright, K. and Monroe, C. (2016). Demonstration of a small programmable quantum computer with atomic qubits. *Nature 2016 536:7614*, **536**, 63–66.

Decelle, A. and Furtlehner, C. (2021). Exact training of restricted boltzmann machines on intrinsically low dimensional data. *Phys. Rev. Lett.*, **127**, 158303.

Deng, L. and Yu, D. (2014). A survey of deep learning: Platforms, applications and emerging research trends. *IEEE access*, **2**, 618–626.

Dennis, E., Kitaev, A., Landahl, A. and Preskill, J. (2002). Topological quantum memory. *Journal of Mathematical Physics*, **43**, 4452–4505.

Deutsch, D. (1985). Quantum theory, the church-turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, **400**, 97–117.

Deutsch, D. and Jozsa, R. (1992). Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, **439**, 553–558.

Ding, J. and Sun, N. (2019). Capacity lower bound for the Ising perceptron. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 816–827, ACM, New York, NY, USA.

Ding, Z. and Lin, L. (2022a). Even shorter quantum circuit for phase estimation on early fault-tolerant quantum computers with applications to ground-state energy estimation. *arXiv*: 2211.11973.

Ding, Z. and Lin, L. (2022b). Even shorter quantum circuit for phase estimation on early fault-tolerant quantum computers with applications to ground-state energy estimation. *arXiv preprint arXiv:2211.11973*.

Ding, Z. and Lin, L. (2023). Even shorter quantum circuit for phase estimation on early fault-tolerant quantum computers with applications to ground-state energy estimation. *PRX Quantum*, **4**, 020331.

Dong, Y., Lin, L. and Tong, Y. (2022a). Ground-state preparation and energy estimation on early fault-tolerant quantum computers via quantum eigenvalue transformation of unitary matrices. *PRX Quantum*, **3**, 040305.

Dong, Y., Lin, L. and Tong, Y. (2022b). Ground-state preparation and energy estimation on early fault-tolerant quantum computers via quantum eigenvalue transformation of unitary matrices. *PRX Quantum*, **3**, 040305.

Du, Y., Hsieh, M.H., Liu, T. and Tao, D. (2020). Expressive power of parametrized quantum circuits. *Phys. Rev. Research*, **2**, 033125.

Díaz, M.G., Sentís, G., Muñoz-Tapia, R. and Sanpera, A. (2021). Volumes of parent hamiltonians for benchmarking quantum simulators.

Echenique, P. and Alonso, J.L. (2007). A mathematical and computational review of hartree–fock scf methods in quantum chemistry. *Molecular Physics*, **105**, 3057–3098.

Egan, L., Debroy, D.M., Noel, C., Risinger, A., Zhu, D., Biswas, D., Newman, M., Li, M., Brown, K.R., Cetina, M. and Monroe, C. (2020). Fault-tolerant operation of a quantum error-correction code.

Elfving, V.E., Broer, B.W., Webber, M., Gavartin, J., Halls, M.D., Lorton, K.P. and Bochevarov, A. (2020a). How will quantum computers provide an industrially relevant computational advantage in quantum chemistry?

Elfving, V.E., Broer, B.W., Webber, M., Gavartin, J., Halls, M.D., Lorton, K.P. and Bochevarov, A. (2020b). How will quantum computers provide an industrially relevant computational advantage in quantum chemistry? *arXiv:2009.12472*.

Fan, F., Cong, W. and Wang, G. (2018). Fuzzy logic interpretation of artificial neural networks. *arXiv preprint arXiv:1807.03215*.

Fan, F., Shan, H., Kalra, M.K., Singh, R., Qian, G., Getzin, M., Teng, Y., Hahn, J. and Wang, G. (2020). Quadratic autoencoder (q-ae) for low-dose ct denoising. *IEEE Transactions on Medical Imaging*, **39**, 2035–2050.

Farhi, E. and Harrow, A.W. (2016). Quantum supremacy through the quantum approximate optimization algorithm. *arXiv preprint arXiv:1602.07674*.

Farhi, E., Goldstone, J. and Gutmann, S. (2014). A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*.

Fellous-Asiani, M. (2021). The resource cost of large scale quantum computing.

Fellous-Asiani, M., Chai, J.H., Whitney, R.S., Auffè ves, A. and Ng, H.K. (2021). Limitations in quantum computing from resource constraints. *PRX Quantum*, **2**.

Fenner, S., Green, A.M., Homer, S., Pruim, R. and Tiu, A. (2009). P-selective sets and weak derandomization for bpp. In *Proceedings of the 24th IEEE Conference on Computational Complexity (CCC '09)*, 249–260.

Feynman, R.P. (1982a). Simulating physics with computers. *International Journal of Theoretical Physics*, **21**, 467–488.

Feynman, R.P. (1982b). Simulating physics with computers. *International Journal of Theoretical Physics*, **21**, 467–488.

Fomichev, S., Hejazi, K., Zini, M.S., Kiser, M., Morales, J.F., Casares, P.A.M., Delgado, A., Huh, J., Voigt, A.C., Mueller, J.E. *et al.* (2023). Initial state preparation for quantum chemistry on quantum computers. *arXiv preprint arXiv:2310.18410*.

Fontanari, J.F. and Meir, R. (1989). Mapping correlated gaussian patterns in a perceptron. *Journal of Physics A*, **22**.

Fowler, A.G. (2012). Proof of finite surface code threshold for matching. *Physical review letters*, **109**, 180502.

Fowler, A.G. and Devitt, S.J. (2012). A bridge to lower overhead quantum computation. *arXiv*: 1209.0510.

Fowler, A.G. and Gidney, C. (2018). Low overhead quantum computation using lattice surgery. *arXiv preprint arXiv:1808.06709*.

Fowler, A.G., Mariantoni, M., Martinis, J.M. and Cleland, A.N. (2012). Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A*, **86**, 032324.

Fraxanet, J., Salamon, T. and Lewenstein, M. (????). The coming decades of quantum simulation. **1000**, arXiv:2204.08905.

Gardner, E. (1988). The space of interactions in neural network models. *Journal of Physics A: Mathematical and General*, **21**, 257–270.

Gardner, E. and Derrida, B. (1988). Optimal storage properties of neural network models. *Journal of Physics A: Mathematical and General*, **21**, 271–284.

Gardner, E. and Derrida, B. (1989a). Three unfinished works on the optimal storage capacity of networks. *Journal of Physics A: Mathematical and General*, **22**, 1983–1994.

Gardner, E. and Derrida, B. (1989b). Three unfinished works on the optimal storage capacity of networks. *Journal of Physics A: Mathematical and General*, **22**, 1983–1994.

Gavriel, J., Herr, D., Shaw, A., Bremner, M.J., Paler, A. and Devitt, S.J. (2022). Transversal injection: A method for direct encoding of ancilla states for non-clifford gates using stabiliser codes. *arXiv preprint arXiv:2211.10046*.

Gili, K., Mauri, M. and Perdomo-Ortiz, A. (2023). Generalization metrics for practical quantum advantage in generative models.

Gilyén, A., Wiebe, N., Berry, D.W. and Childs, A.M. (2015). Ground-state preparation and energy estimation on early fault-tolerant quantum computers via quantum eigenvalue transformation of unitary matrices. *Phys. Rev. A*, **92**, 042303.

Giurgica-Tiron, T., Hindy, Y., LaRose, R., Mari, A. and Zeng, W.J. (2020). Digital zero noise extrapolation for quantum error mitigation. 306–316, IEEE.

Giurgica-Tiron, T., Johri, S., Kerenidis, I., Nguyen, J., Pisenti, N., Prakash, A., Sosnova, K., Wright, K. and Zeng, W. (2022a). Low-depth amplitude estimation on a trapped-ion quantum computer. *Physical Review Research*, **4**, 033034.

Giurgica-Tiron, T., Kerenidis, I., Labib, F., Prakash, A. and Zeng, W. (2022b). Low depth algorithms for quantum amplitude estimation. *Quantum*, **6**, 745.

Goings, J.J., White, A., Lee, J., Tautermann, C.S., Degroote, M., Gidney, C., Shiozaki, T., Babbush, R. and Rubin, N.C. (2022a). Reliably assessing the electronic structure of cytochrome p450 on today's classical computers and tomorrow's quantum computers. *Proceedings of the National Academy of Sciences*, **119**.

Goings, J.J., White, A., Lee, J., Tautermann, C.S., Degroote, M., Gidney, C., Shiozaki, T., Babbush, R. and Rubin, N.C. (2022b). Reliably assessing the electronic structure of cytochrome p450 on today's classical computers and tomorrow's quantum computers. *Proceedings of the National Academy of Sciences*, **119**.

Goings, J.J., White, A., Lee, J., Tautermann, C.S., Degroote, M., Gidney, C., Shiozaki, T., Babbush, R. and Rubin, N.C. (2022c). Reliably assessing the electronic structure of cytochrome p450 on today's classical computers and tomorrow's quantum computers. *Proceedings of the National Academy of Sciences*, **119**, e2203533119.

Gonthier, J.F., Radin, M.D., Buda, C., Doskocil, E.J., Abuan, C.M. and Romero, J. (2020). Identifying challenges towards practical quantum advantage through resource estimation: the measurement roadblock in the variational quantum eigensolver.

Gonthier, J.F., Radin, M.D., Buda, C., Doskocil, E.J., Abuan, C.M. and Romero, J. (2022a). Measurements as a roadblock to near-term practical quantum advantage in chemistry: Resource analysis. *Physical Review Research*, **4**.

Gonthier, J.F., Radin, M.D., Buda, C., Doskocil, E.J., Abuan, C.M. and Romero, J. (2022b). Measurements as a roadblock to near-term practical quantum advantage in chemistry: resource analysis. *Physical Review Research*, **4**, 033154.

Gottesman, D. (1997). Stabilizer codes and quantum error correction.

Graner, B., Cairncross, W.B., Gurevich, Y.V., Hutzler, N.R., Kirilov, E., O'Leary, B.R., Rajendran, S., Weiner, I. and Demille, D. (2019). New limit on the permanent electric dipole moment of hg199. *Physical Review Letters*, **123**, 030000.

Grant, E., Wossnig, L., Ostaszewski, M., Benedetti, M. and Fitzsimons, J.F. (2018). Hierarchical quantum classifiers. *npj Quantum Information*, **4**, 1–9.

Gratsea, A. and Huembeli, P. (2021). Exploring quantum perceptron and quantum neural network structures with a teacher-student scheme.

Gratsea, A. and Huembeli, P. (2022). The effect of the processing and measurement operators on the expressive power of quantum models.

Gratsea, A., Kasper, V. and Lewenstein, M. (2021). Storage properties of a quantum perceptron.

Gratsea, K., Sun, C. and Johnson, P.D. (2022). When to reject a ground state preparation algorithm.

Gratsea, K., Kottmann, J.S., Johnson, P.D. and Kunitsa, A.A. (2024). Comparing classical and quantum ground state preparation heuristics.

Grimsley, H.R., Economou, S.E., Barnes, E. and Mayhall, N.J. (2019). An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nature Communications*, **10**.

Grover, L.K. (1996). A fast quantum mechanical algorithm for database search. *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, 212–219.

Grover, L.K. (1997). Quantum mechanics helps in searching for a needle in a haystack. *Physical Review Letters*, **79**, 325.

Gu, L., Zhou, F. and Yang, L. (2020). Towards the representational power of restricted Boltzmann machines. *Neurocomputing*, **415**, 358–367.

Haferkamp, J., Faist, P., Kothakonda, N.B.T., Eisert, J. and Halpern, N.Y. (2022). Linear growth of quantum circuit complexity. *Nature Physics*, **18**, 528–532.

Hanneke, D., Home, J.P., Jost, J.D., Amini, J.M., Leibfried, D. and Wineland, D.J. (2009). Realization of a programmable two-qubit quantum processor. *Nature Physics 2009 6:1*, **6**, 13–16.

Harrison, R.J., Fann, G.I., Yanai, T., Gan, Z. and Beylkin, G. (2004). Multiresolution quantum chemistry: Basic theory and initial applications. *The Journal of Chemical Physics*, **121**, 11587–11598.

Harrison, R.J., Beylkin, G., Bischoff, F.A., Calvin, J.A., Fann, G.I., Fosso-Tande, J., Galindo, D., Hammond, J.R., Hartman-Baker, R., Hill, J.C., Jia, J., Kottmann, J.S., Ou, M.J.Y., Pei, J., Ratcliff, L.E., Reuter, M.G., Richie-Halford, A.C., Romero, N.A., Sekino, H., Shelton, W.A., Sundahl, B.E., Thornton, W.S., Valeev, E.F., Vázquez-Mayagoitia, Á ., Vence, N., Yanai, T. and Yokoi, Y. (2016). MADNESS: A multiresolution, adaptive numerical environment for scientific simulation. *SIAM Journal on Scientific Computing*, **38**, S123–S142.

Havlíček, V., Córcoles, A.D., Temme, K., Harrow, A.W., Kandala, A., Chow, J.M. and Gambetta, J.M. (2019). Supervised learning with quantum-enhanced feature spaces. *Nature*, **567**, 209–212.

Hibat-Allah, M., Mauri, M., Carrasquilla, J. and Perdomo-Ortiz, A. (2023). A framework for demonstrating practical quantum advantage: Racing quantum against classical generative models.

Hinton, G., Vinyals, O. and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci.*, **79**, 2554–2558.

Hopfield, J.J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences*, **81**, 3088–3092.

Hopfield, J.J. and Tank, D.W. (1986). Computing with neural circuits: A model. *Science*, **233**, 625–633.

Huggins, W.J., McArdle, S., O'Brien, T.E., Lee, J., Rubin, N.C., Boixo, S., Whaley, K.B., Babbush, R. and McClean, J.R. (2021). Virtual distillation for quantum error mitigation. *Physical Review X*, **11**, 041036.

Hutchings, M.D., Hertzberg, J.B., Liu, Y., Bronn, N.T., Keefe, G.A., Brink, M., Chow, J.M. and Plourde, B.L. (2017). Tunable superconducting qubits with flux-independent coherence. *Physical Review Applied*, **8**, 044003.

Jerbi, S., Fiderer, L.J., Nautrup, H.P., Kübler, J.M., Briegel, H.J. and Dunjko, V. (2021). Quantum machine learning beyond kernel methods.

Johnson, P.D., Kunitsa, A.A., Gonthier, J.F., Radin, M.D., Buda, C., Doskocil, E.J., Abuan, C.M. and Romero, J. (2022). Reducing the cost of energy estimation in the variational quantum eigensolver algorithm with robust amplitude estimation. *arXiv preprint arXiv:2203.07275*.

Jones, C. (2013). Low-overhead constructions for the fault-tolerant toffoli gate. *Phys. Rev. A*, **87**, 022328.

Kak, S. (1995). Quantum neural computing. *Advances in Imaging and Electron Physics*, **94**, 259–313.

Kasper, V., González-Cuadra, D., Hegde, A., Xia, A., Dauphin, A., Huber, F., Tiemann, E., Lewenstein, M., Jendrzejewski, F. and Hauke, P. (2020). Universal quantum computation and quantum error correction with ultracold atomic mixtures.

Katabarwa, A., Kunitsa, A., Peropadre, B. and Johnson, P. (2021). Reducing runtime and error in vqe using deeper and noisier quantum circuits. *arXiv preprint arXiv:2110.10664*.

Katabarwa, A., Gratsea, K., Caesura, A. and Johnson, P.D. (2023). Early fault-tolerant quantum computing.

Kechedzhi, K., Isakov, S.V., Mandrà, S., Villalonga, B., Mi, X., Boixo, S. and Smelyanskiy, V. (2023). Effective quantum volume, fidelity and computational cost of noisy quantum processing experiments.

Kim, I.H., Liu, Y.H., Pallister, S., Pol, W., Roberts, S. and Lee, E. (2022a). Fault-tolerant resource estimate for quantum chemical simulations: Case study on li-ion battery electrolyte molecules. *Physical Review Research*, **4**, 023019.

Kim, I.H., Liu, Y.H., Pallister, S., Pol, W., Roberts, S. and Lee, E. (2022b). Fault-tolerant resource estimate for quantum chemical simulations: Case study on li-ion battery electrolyte molecules. *Physical Review Research*, **4**.

Kim, I.H., Liu, Y.H., Pallister, S., Pol, W., Roberts, S. and Lee, E. (2022c). Fault-tolerant resource estimate for quantum chemical simulations: Case study on li-ion battery electrolyte molecules. *Physical Review Research*, **4**.

Kim, Y., Eddins, A., Anand, S., Wei, K., Berg, E., Rosenblatt, S., Nayfeh, H., Wu, Y., Zaletel, M., Temme, K. and Kandala, A. (2023a).

Evidence for the utility of quantum computing before fault tolerance. *Nature*, **618**, 500–505.

Kim, Y., Eddins, A., Anand, S., Wei, K.X., van den Berg, E., Rosenblatt, S., Nayfeh, H., Wu, Y., Zaletel, M., Temme, K. and Kandala, A. (2023b). Evidence for the utility of quantum computing before fault tolerance. *Nature 2023 618:7965*, **618**, 500–505.

Kimmel, S., Low, G.H. and Yoder, T.J. (2015). Robust calibration of a universal single-qubit gate set via robust phase estimation. *Physical Review A*, **92**, 062315.

Kirby, W., Motta, M. and Mezzacapo, A. (2023). Exact and efficient lanczos method on a quantum computer. *Quantum*, **7**, 1018.

Kitaev, A.Y. (2003). Fault-tolerant quantum computation by anyons. *Annals of Physics*, **303**, 2–30.

Klymko, K., Mejuto-Zaera, C., Cotton, S.J., Wudarski, F., Urbanek, M., Hait, D., Head-Gordon, M., Whaley, K.B., Moussa, J., Wiebe, N. *et al.* (2022). Real-time evolution for ultracompact hamiltonian eigenstates on quantum hardware. *PRX Quantum*, **3**, 020323.

Knill, E. and Laflamme, R. (1996). Concatenated quantum codes.

Knill, E., Ortiz, G. and Somma, R.D. (2007). Optimal quantum measurements of expectation values of observables. *Phys. Rev. A*, **75**, 012328.

Kohn, W. (1999). Nobel lecture: Electronic structure of matter—wave functions and density functionals. *Rev. Mod. Phys.*, **71**, 1253–1266.

Kohring, G. (1990). Neural networks with many-neuron interactions. *Journal de Physique*, **51**, 145–155.

Kottmann, J.S. (2022). Molecular quantum circuit design: A graph-based approach.

Kottmann, J.S. and Aspuru-Guzik, A. (2022a). Optimized low-depth quantum circuits for molecular electronic structure using a separable-pair approximation. *Physical Review A*, **105**.

Kottmann, J.S. and Aspuru-Guzik, A. (2022b). Optimized low-depth quantum circuits for molecular electronic structure using a separable-pair approximation. *Physical Review A*, **105**.

Kottmann, J.S., Bischoff, F.A. and Valeev, E.F. (2020). Direct determination of optimal pair-natural orbitals in a real-space representation: The second-order Moller–Plesset energy. *The Journal of Chemical Physics*, **152**, 074105.

Kottmann, J.S., Alperin-Lea, S., Tamayo-Mendoza, T., Cervera-Lierta, A., Lavigne, C., Yen, T.C., Verteletskyi, V., Schleich, P., Anand, A., Degroote, M., Chaney, S., Kesibi, M., Curnow, N.G., Solo, B., Tsilimigkounakis, G., Zendejas-Morales, C., Izmaylov, A.F. and Aspuru-Guzik, A. (2021a). TEQUILA: a platform for rapid development of quantum algorithms. *Quantum Science and Technology*, **6**, 024009.

Kottmann, J.S., Schleich, P., Tamayo-Mendoza, T. and Aspuru-Guzik, A. (2021b). Reducing qubit requirements while maintaining numerical precision for the variational quantum eigensolver: A basis-set-free approach. *The Journal of Physical Chemistry Letters*, **12**, 663–673.

Kovalev, A.A. and Pryadko, L.P. (2013). Fault tolerance of quantum low-density parity check codes with sublinear distance scaling. *Physical Review A - Atomic, Molecular, and Optical Physics*, **87**, 020304.

Krauth, W. and Mézard, M. (1989). Storage capacity of memory networks with binary couplings. *Journal de Physique*, **50**, 3057–3066.

Kshirsagar, R., Katabarwa, A. and Johnson, P.D. (2022). On proving the robustness of algorithms for early fault-tolerant quantum computers. *arXiv preprint arXiv:2209.11322*.

Laflamme, R., Miquel, C., Paz, J.P. and Zurek, W.H. (1996). Perfect quantum error correcting code. *Physical Review Letters*, **77**, 198.

Le Roux, N. and Bengio, Y. (2008). Representational power of restricted boltzmann machines and deep belief networks. *Neural Computation*, **20**, 1631–1649.

LeCun, Y. (1998). Efficient backprop. *Neural Networks: Tricks of the Trade*, 9–48.

LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86**, 2278–2324.

Lee, J., Berry, D.W., Gidney, C., Huggins, W.J., McClean, J.R., Wiebe, N. and Babbush, R. (2021). Even more efficient quantum computations of chemistry through tensor hypercontraction. *PRX Quantum*, **2**, 030305.

Lee, S., Lee, J., Zhai, H., Tong, Y., Dalzell, A.M., Kumar, A., Helms, P., Gray, J., Cui, Z.H., Liu, W., Kastoryano, M., Babbush, R., Preskill, J., Reichman, D.R., Campbell, E.T., Valeev, E.F., Lin, L. and Chan, G.K.L. (2022). Is there evidence for exponential quantum advantage in quantum chemistry?

Leung, P.H. and Brown, K.R. (2018). Entangling an arbitrary pair of qubits in a long ion crystal. *Physical Review A*, **98**, 032318.

Lewenstein, M. (1994). Quantum perceptrons. *Journal of Modern Optics*, **41**, 2491–2501.

Lewenstein, M., Gratsea, A., Riera-Campeny, A., Aloy, A., Kasper, V. and Sanpera, A. (2021). Storage capacity and learning capability of quantum neural networks. *Quantum Science and Technology*, **6**, 045002.

Li, H., Ni, H. and Ying, L. (2023). On low-depth quantum algorithms for robust multiple-phase estimation. *arXiv preprint arXiv:2303.08099*.

Liang, Q., Kang, M., Li, M. and Nam, Y. (2023a). Pulse optimization for high-precision motional-mode characterization in trapped-ion quantum computers.

Liang, Q., Zhou, Y., Dalal, A. and Johnson, P.D. (2023b). Modeling the performance of early fault-tolerant quantum algorithms. *arXiv preprint arXiv:2306.17235*.

Lin, L. and Tong, Y. (2020a). Near-optimal ground state preparation. *Quantum*, **4**, 372.

Lin, L. and Tong, Y. (2020b). Near-optimal ground state preparation. *Quantum*, **4**, 372.

Lin, L. and Tong, Y. (2020c). Near-optimal ground state preparation. *Quantum*, **4**, 372.

Lin, L. and Tong, Y. (2022a). Heisenberg-limited ground-state energy estimation for early fault-tolerant quantum computers. *PRX Quantum*, **3**, 010318.

Lin, L. and Tong, Y. (2022b). Heisenberg-limited ground-state energy estimation for early fault-tolerant quantum computers. *PRX Quantum*, **3**, 010318.

Litinski, D. (2019). Magic state distillation: Not as costly as you think. *Quantum*, **3**, 205.

Liu, J.G. and Wang, L. (2018a). Differentiable learning of quantum circuit born machines. *Physical Review A*, **98**.

Liu, J.G. and Wang, L. (2018b). Differentiable learning of quantum circuit born machines. *Physical Review A*, **98**, 062324.

Lloyd, S. (1993). Quantum-mechanical computers and uncomputability. *Phys. Rev. Lett.*, **71**, 943–946.

Lykov, D., Wurtz, J., Poole, C., Saffman, M., Noel, T. and Alexeev, Y. (2023). Sampling frequency thresholds for the quantum advantage of the quantum approximate optimization algorithm. *npj Quantum Information 2023 9:1*, **9**, 1–10.

Mangini, S., Tacchino, F., Gerace, D., Macchiavello, C. and Bajoni, D. (2020). Quantum computing model of an artificial neuron with continuously valued input data. *Machine Learning: Science and Technology*, **1**, 045008.

Mayer, I. (2003). *Simple theorems, proofs, and derivations in quantum chemistry*. Springer Science & Business Media.

McClean, J. (????). Integrating over the unitary group. Accessed: OCTOBER 14, 2015.

McClean, J.R., Romero, J., Babbush, R. and Aspuru-Guzik, A. (2016). The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, **18**, 023023.

McClean, J.R., Rubin, N.C., Sung, K.J., Kivlichan, I.D., Bonet-Monroig, X., Cao, Y., Dai, C., Fried, E.S., Gidney, C., Gimby, B., Gokhale, P., Häner, T., Hardikar, T., Havlíček, V., Higgott, O., Huang, C., Izaac, J., Jiang, Z., Liu, X., McArdle, S., Neeley, M., O′Brien, T., O′Gorman, B., Ozfidan, I., Radin, M.D., Romero, J., Sawaya, N.P.D., Senjean, B., Setia, K., Sim, S., Steiger, D.S., Steudtner, M., Sun, Q., Sun, W., Wang, D., Zhang, F. and Babbush, R. (2020). Openfermion: the electronic structure package for quantum computers. *Quantum Science and Technology*, **5**, 034014.

McClelland, J.L., Rumelhart, D.E. and PDP Research Group, eds. (1986). *Parallel Distributed Processing. Volume 2: Psychological and Biological Models*. MIT Press, Cambridge, MA.

Meinhardt, N., Neumann, N.M.P. and Phillipson, F. (2020). *Quantum Hopfield Neural Networks*. Springer International Publishing.

Mezard, M., Parisi, G. and Virasoro, M. (1986). *Spin Glass Theory and Beyond*. WORLD SCIENTIFIC.

Minsky, M. and Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, MA, USA.

Minsky, M. and Papert, S. (1972). *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, Cambridge MA.

Minsky, M. and Papert, S. (1988). *Perceptrons*. MIT press, 2nd edn.

Mitarai, K., Negoro, M., Kitagawa, M. and Fujii, K. (2018). Quantum circuit learning. *Physical Review A*, **98**, 032309.

Monroe, C., Meekhof, D.M., King, B.E., Itano, W.M. and Wineland, D.J. (1995). Demonstration of a fundamental quantum logic gate. *Physical Review Letters*, **75**, 4714.

Montanaro, A. (2016). Quantum algorithms: an overview. *npj Quantum Information*, **2**, 15023.

Motta, M., Genovese, C., Ma, F., Cui, Z.H., Sawaya, R., Chan, G.K.L., Chepiga, N., Helms, P., Jimé nez-Hoyos, C., Millis, A.J., Ray, U., Ronca, E., Shi, H., Sorella, S., Stoudenmire, E.M., White, S.R. and and, S.Z. (2020). Ground-state properties of the hydrogen chain: Dimerization, insulator-to-metal transition, and magnetic phases. *Physical Review X*, **10**.

Motta, M., Ye, E., McClean, J.R., Li, Z., Minnich, A.J., Babbush, R. and Chan, G.K.L. (2021). Low rank representations for quantum simulation of electronic structure. *npj Quantum Information*, **7**.

Müller, B., Reinhardt, J. and Strickland, M.T. (1995). *Neural Networks*. Physics of Neural Networks, Springer Berlin Heidelberg, Berlin, Heidelberg.

Nahum, A., Vijay, S. and Haah, J. (2018). Operator spreading in random unitary circuits. *Physical Review X*, **8**.

Nielsen, M.A. and Chuang, I.L. (2011). Quantum computation and quantum information: 10th anniversary edition.

Nishimori, H. (2001). *Statistical Physics of Spin Glasses and Information Processing*. Oxford University Press.

Parrish, R.M. and McMahon, P.L. (2019). Quantum filter diagonalization: Quantum eigendecomposition without full quantum phase estimation. *arXiv preprint arXiv:1909.08925*.

Patel, K.N., Markov, I.L. and Hayes, J.P. (2003). Efficient synthesis of linear reversible circuits. *arXiv preprint quant-ph/0302002*.

Pathak, S., Russo, A.E., Seritan, S.K. and Baczewski, A.D. (2023). Quantifying t-gate-count improvements for ground-state-energy estimation with near-optimal state preparation. *Physical Review A*, **107**.

Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.H., Zhou, X.Q., Love, P.J., Aspuru-Guzik, A. and O'Brien, J.L. (2014a). A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, **5**.

Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.H., Zhou, X.Q., Love, P.J., Aspuru-Guzik, A. and O'Brien, J.L. (2014b). A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, **5**, 4213.

Piveteau, C., Sutter, D., Bravyi, S., Gambetta, J.M. and Temme, K. (2021). Error mitigation for universal gates on encoded qubits. *Phys. Rev. Lett.*, **127**, 200505.

Poggio, T., Mhaskar, H., Rosasco, L., Miranda, B. and Liao, Q. (2017). Why and when can deep – but not shallow – networks avoid the curse of dimensionality: a review.

Postler, L., Heussen, S., Pogorelov, I., Rispler, M., Feldker, T., Meth, M., Marciniak, C.D., Stricker, R., Ringbauer, M., Blatt, R., Schindler, P., Müller, M. and Monz, T. (2022). Demonstration of fault-tolerant universal quantum gate operations. *Nature*, **605**, 675–680.

Poulin, D., Kitaev, A., Steiger, D.S., Hastings, M.B. and Troyer, M. (2018). Quantum algorithm for spectral measurement with a lower gate count. *Physical review letters*, **121**, 010501.

Pozas-Kerstjens, A., Muñoz-Gil, G., Ángel García-March, M., Acín, A., Lewenstein, M. and Grzybowski, P.R. (2021). Efficient training of energy-based models via spin-glass control. *Mach. Learn.: Sci. Techn.*, **2**, 025026.

Preskill, J. (2018). Quantum computing in the nisq era and beyond. *Quantum*, **2**, 79.

Pérez-Salinas, A., Cervera-Lierta, A., Gil-Fuster, E. and Latorre, J.I. (2020). Data re-uploading for a universal quantum classifier. *Quantum*, **4**, 226.

Pérez-Salinas, A., Cruz-Martinez, J., Alhajri, A.A. and Carrazza, S. (2021a). Determining the proton content with a quantum computer.

Pérez-Salinas, A., López-Núñez, D., García-Sáez, A., Forn-Díaz, P. and Latorre, J.I. (2021b). One qubit as a universal approximant.

Qiskit contributors (2023). Qiskit: An open-source framework for quantum computing.

Raussendorf, R., Harrington, J. and Goyal, K. (2006). A fault-tolerant one-way quantum computer. *Annals of Physics*, **321**, 2242–2270.

Rebentrost, P., Bromley, T.R., Weedbrook, C. and Lloyd, S. (2018). Quantum hopfield neural network. *Physical Review A*, **98**.

Ringbauer, M., Meth, M., Postler, L., Stricker, R., Blatt, R., Schindler, P. and Monz, T. (2021). A universal qudit quantum processor with trapped ions.

Roberts, D.A., Stanford, D. and Streicher, A. (2018). Operator growth in the SYK model. *Journal of High Energy Physics*, **2018**.

Rojas, R. (1996). *Neural Networks - A Systematic Introduction*. Springer-Verlag, Berlin.

Rosenblatt, F. (1957). *The Perceptron, a Perceiving and Recognizing Automaton Project Para*. Report: Cornell Aeronautical Laboratory, Cornell Aeronautical Laboratory.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, **65**, 386–408.

Ross, N.J. and Selinger, P. (2016). Optimal ancilla-free clifford+t approximation of z-rotations.

Rumelhart, D.E., McClelland, J.L. and PDP Research Group, eds. (1986). *Parallel Distributed Processing. Volume 1: Foundations*. MIT Press, Cambridge, MA.

Saad, Y. (2011). *Numerical methods for large eigenvalue problems: revised edition*. SIAM.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, **61**, 85–117.

Schuld, M. (2021a). Quantum machine learning models are kernel methods.

Schuld, M. (2021b). Supervised quantum machine learning models are kernel methods.

Schuld, M. and Killoran, N. (2022a). Is quantum advantage the right goal for quantum machine learning? *Phys. Rev. X Quantum*, **3**, 030101.

Schuld, M. and Killoran, N. (2022b). Is quantum advantage the right goal for quantum machine learning?

Schuld, M., Sinayskiy, I. and Petruccione, F. (2014). The quest for a Quantum Neural Network. *Quantum Information Processing*, **13**, 2567–2586.

Schuld, M., Sinayskiy, I. and Petruccione, F. (2015a). Simulating a perceptron on a quantum computer. *Physics Letters A*, **379**, 660–663.

Schuld, M., Sinayskiy, I. and Petruccione, F. (2015b). Simulating a perceptron on a quantum computer. *Physics Letters A*, **379**, 660–663.

Schuld, M., Fingerhuth, M. and Petruccione, F. (2018). Implementing a distance-based classifier with a quantum interference circuit. *Europhysics Letters*, **120**, 60002.

Schuld, M., Bocharov, A., Svore, K.M. and Wiebe, N. (2020a). Circuit-centric quantum classifiers. *Physical Review A*, **101**.

Schuld, M., Sweke, R. and Meyer, J.J. (2020b). The effect of data encoding on the expressive power of variational quantum machine learning models.

Schönhage, A. and Strassen, V. (1971). Schnelle multiplikation großer zahlen. In *Computational Complexity*, vol. 1, 117–123.

Sharma, K., Cerezo, M., Cincio, L. and Coles, P.J. (2020). Trainability of dissipative perceptron-based quantum neural networks.

Shaydulin, R., Li, C., Chakrabarti, S., DeCross, M., Herman, D., Kumar, N., Larson, J., Lykov, D., Minssen, P., Sun, Y. *et al.* (2023). Evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem. *arXiv preprint arXiv:2308.02342*.

Shen, H., Zhang, P., You, Y.Z. and Zhai, H. (2020). Information scrambling in quantum neural networks. *Phys. Rev. Lett.*, **124**, 200504.

Sherrill, C.D. and Schaefer III, H.F. (1999). The configuration interaction method: Advances in highly correlated approaches. In *Advances in quantum chemistry*, vol. 34, 143–269, Elsevier.

Sherrington, D. and Kirkpatrick, S. (1975). Solvable Model of a Spin-Glass. *Physical Review Letters*, **35**, 1792–1796.

Shi, X.F. (2022). Quantum logic and entanglement by neutral rydberg atoms: methods and fidelity. *Quantum Science and Technology*, **7**, 023002.

Shor, P. (????). Fault-tolerant quantum computation. *Proceedings of 37th Conference on Foundations of Computer Science*, 56–65.

Shor, P.W. (1995). Scheme for reducing decoherence in quantum computer memory. *Physical Review A*, **52**, R2493.

Shor, P.W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, **26**, 1484–1509.

Sim, S., Johnson, P.D. and Aspuru-Guzik, A. (2019a). Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, **2**, 1900070.

Sim, S., Johnson, P.D. and Aspuru-Guzik, A. (2019b). Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, **2**, 1900070.

Sompolinsky, H. (1987). The theory of neural networks: The hebb rule and beyond. In J.L. van Hemmen and I. Morgenstern, eds., *Heidelberg Colloquium on Glassy Dynamics*, 485–527, Springer Berlin Heidelberg, Berlin, Heidelberg.

Stair, N.H. and Evangelista, F.A. (2020). Exploring Hilbert space on a budget: Novel benchmark set and performance metric for testing electronic structure methods in the regime of strong correlation. *The Journal of Chemical Physics*, **153**, 104108.

Steane, A.M. (1997). Active stabilization, quantum computation, and quantum state synthesis. *Physical Review Letters*, **78**, 2252.

Su, Y., Berry, D.W., Wiebe, N., Rubin, N. and Babbush, R. (2021). Fault-tolerant quantum simulations of chemistry in first quantization. *PRX Quantum*, **2**, 040332.

Sun, Q., Berkelbach, T.C., Blunt, N.S., Booth, G.H., Guo, S., Li, Z., Liu, J., McClain, J.D., Sayfutyarova, E.R., Sharma, S., Wouters, S. and Chan, G.K.L. (2018). Pyscf: the python-based simulations of chemistry framework. *WIREs Computational Molecular Science*, **8**, e1340.

Suzuki, Y., Kawase, Y., Masumura, Y., Hiraga, Y., Nakadai, M., Chen, J., Nakanishi, K.M., Mitarai, K., Imai, R., Tamiya, S., Yamamoto, T., Yan, T., Kawakubo, T., Nakagawa, Y.O., Ibe, Y., Zhang, Y., Yamashita, H., Yoshimura, H., Hayashi, A. and Fujii, K. (2021). Qulacs: a fast and versatile quantum circuit simulator for research purpose. *Quantum*, **5**, 559.

Suzuki, Y., Endo, S., Fujii, K. and Tokunaga, Y. (2022). Quantum error mitigation as a universal error reduction technique: Applications from the nisq to the fault-tolerant quantum computing eras. *PRX Quantum*, **3**, 010345.

Sweke, R., Wilde, F., Meyer, J., Schuld, M., Faehrmann, P.K., Meynard-Piganeau, B. and Eisert, J. (2020). Stochastic gradient descent for hybrid quantum-classical optimization. *Quantum*, **4**, 314.

Tacchino, F., Macchiavello, C., Gerace, D. and Bajoni, D. (2019). An artificial neuron implemented on an actual quantum processor. *npj Quantum Information*, **5**, 26.

Tacchino, F., Barkoutsos, P., Macchiavello, C., Tavernelli, I., Gerace, D. and Bajoni, D. (2020a). Quantum implementation of an artificial feed-forward neural network. *Quantum Science and Technology*, **5**, 044010.

Tacchino, F., Barkoutsos, P.K., Macchiavello, C., Gerace, D., Tavernelli, I. and Bajoni, D. (2020b). Variational learning for quantum artificial neural networks. *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 130–136.

Temme, K., Bravyi, S. and Gambetta, J.M. (2017). Error mitigation for short-depth quantum circuits. *Physical Review Letters*, **119**, 180509.

Tilly, J., Chen, H., Cao, S., Picozzi, D., Setia, K., Li, Y., Grant, E., Wossnig, L., Rungger, I., Booth, G.H. and Tennyson, J. (2022). The variational quantum eigensolver: A review of methods and best practices. *Physics Reports*, **986**, 1–128.

Tindall, J., Fishman, M., Stoudenmire, E.M. and Sels, D. (2023). Efficient tensor network simulation of ibm's eagle kicked ising experiment.

Tishby, N. (????).

Torrey, L. and Taylor, M.E. (2013). Teaching on a budget: Agents advising agents in reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Torrontegui, E. and García-Ripoll, J.J. (2019). Unitary quantum perceptron as efficient universal approximator. *Europhysics Letters*, **125**, 30004.

Torta, P., Mbeng, G.B., Baldassi, C., Zecchina, R. and Santoro, G.E. (2021). Quantum approximate optimization algorithm applied to the binary perceptron.

Tranter, A., Love, P.J., Mintert, F., Wiebe, N. and Coveney, P.V. (2019). Ordering of trotterization: Impact on errors in quantum simulation of electronic structure. *Entropy*, **21**, 1218.

Tubman, N.M., Mejuto-Zaera, C., Epstein, J.M., Hait, D., Levine, D.S., Huggins, W., Jiang, Z., McClean, J.R., Babbush, R., Head-Gordon, M. and Whaley, K.B. (2018). Postponing the orthogonality catastrophe: efficient state preparation for electronic structure simulations on quantum devices.

van den Berg, E., Minev, Z.K., Kandala, A. and Temme, K. (2023). Probabilistic error cancellation with sparse pauli–lindblad models on noisy quantum processors. *Nature Physics*, **19**, 1116–1121.

Ventura, D. and Martinez, T. (1998). Quantum associative memory with exponential capacity. **1**, 509–513 vol.1.

Ventura, D. and Martinez, T. (2000). Quantum associative memory. *Information Sciences*, **124**, 273–296.

Vinyals, O., Babuschkin, I., Czarnecki, W.M., Mathieu, M., Dudzik, A., Chung, J., Choi, D.H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J.P., Jaderberg, M., Vezhnevets, A.S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T.L., Gulcehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Kavukcuoglu, K., Hassabis, D., Apps, C. and Silver, D. (2019).

Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, **575**, 350–354.

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P. and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, **17**, 261–272.

von Burg, V., Low, G.H., Häner, T., Steiger, D.S., Reiher, M., Roetteler, M. and Troyer, M. (2021). Quantum computing enhanced computational catalysis. *Physical Review Research*, **3**.

Wan, K., Berta, M. and Campbell, E.T. (2022). Randomized quantum algorithm for statistical phase estimation. *Physical Review Letters*, **129**, 030503.

Wang, D., Higgott, O. and Brierley, S. (2019). Accelerated variational quantum eigensolver. *Physical review letters*, **122**, 140504.

Wang, D.S., Fowler, A.G. and Hollenberg, L.C. (2011). Surface code quantum computing with error rates over 1%. *Physical Review A*, **83**, 020302.

Wang, G., Koh, D.E., Johnson, P.D. and Cao, Y. (2021). Minimizing estimation runtime on noisy quantum computers. *PRX Quantum*, **2**, 010346.

Wang, G., França, D.S., Zhang, R., Zhu, S. and Johnson, P.D. (2022a). Quantum algorithm for ground state energy estimation using circuit depth with exponentially improved dependence on precision.

Wang, G., Sim, S. and Johnson, P.D. (2022b). State preparation boosters for early fault-tolerant quantum computation. *Quantum*, **6**, 829.

Wang, G., Sim, S. and Johnson, P.D. (2022c). State preparation boosters for early fault-tolerant quantum computation. *Quantum*, **6**, 829.

Wang, G., Stilck-França, D., Zhang, R., Zhu, S. and Johnson, P.D. (2022d). Quantum algorithm for ground state energy estimation using circuit depth with exponentially improved dependence on precision.

Wang, G., França, D.S., Zhang, R., Zhu, S. and Johnson, P.D. (2023a). Quantum algorithm for ground state energy estimation using circuit depth with exponentially improved dependence on precision. *Quantum*, **7**, 1167.

Wang, G., França, D.S., Rendon, G. and Johnson, P.D. (2023b). Faster ground state energy estimation on early fault-tolerant quantum computers via rejection sampling.

Weggemans, J.R., Urech, A., Rausch, A., Spreeuw, R., Boucherie, R., Schreck, F., Schoutens, K., Minář, J. and Speelman, F. (2021). Solving correlation clustering with qaoa and a rydberg qudit system: a full-stack approach.

Werbos, P.J. (1974). The activation function in the perceptron algorithm. *Proceedings of the 1974 Symposium on Pattern Recognition and Signal Processing*, 25–32.

Wiebe, N., Kapoor, A. and Svore, K.M. (2016). Quantum perceptron models.

Wintersperger, K., Dommert, F., Ehmer, T., Hoursanov, A., Klepsch, J., Mauerer, W., Reuber, G., Strohm, T., Yin, M. and Luber, S. (2023). Neutral atom quantum computing hardware: performance and end-user perspective. *EPJ Quantum Technology 2023 10:1*, **10**, 1–26.

Wittek, P. (2014). *Quantum machine learning: what quantum computing means to data mining*. Academic Press.

Wouters, S., Poelmans, W., Ayers, P.W. and Van Neck, D. (2014). Chemps2: A free open-source spin-adapted implementation of the density matrix renormalization group for ab initio quantum chemistry. *Computer Physics Communications*, **185**, 1501–1514.

Wright, L.G. and McMahon, P.L. (2020). The Capacity of Quantum Neural Networks. In *Conference on Lasers and Electro-Optics*, JM4G.5, OSA, Washington, D.C.

Wu, Y., Zhang, P. and Zhai, H. (2021). Scrambling ability of quantum neural network architectures. *Physical Review Research*, **3**.

Xiao, X., Freericks, J.K. and Kemper, A.F. (2023). Robust measurement of wave function topology on NISQ quantum computers. *Quantum*, **7**, 987.

Xue, X., Russ, M., Samkharadze, N., Undseth, B., Sammak, A., Scappucci, G. and Vandersypen, L.M.K. (2022). Quantum logic with spin qubits crossing the surface code threshold. *Nature*, **601**, 343–347.

Yan, S., Qi, H. and Cui, W. (2020). Nonlinear quantum neuron: A fundamental building block for quantum neural networks. *Physical Review A*, **102**.

Younes, L. (1996). Synchronous boltzmann machines can be universal approximators. *Applied Mathematics Letters*, **9**, 109–113.

Zalka, C. (1996). Threshold estimate for fault tolerant quantum computation. *arXiv preprint quant-ph/9612028*.

Zapata AI (2023). Orquestra workflow sdk.

Zhai, H., Larsson, H.R., Lee, S., Cui, Z.H., Zhu, T., Sun, C., Peng, L., Peng, R., Liao, K., Tölle, J. *et al.* (2023). Block2: a comprehensive open source framework to develop and apply state-of-the-art dmrg algorithms in electronic structure and beyond. *arXiv preprint arXiv:2310.03920*.

Zhang, R., Wang, G. and Johnson, P. (2021). Computing ground state properties with early fault-tolerant quantum computers. *Quantum*, **6**, 761.

Zhang, R., Wang, G. and Johnson, P. (2022). Computing ground state properties with early fault-tolerant quantum computers. *Quantum*, **6**, 761.

Życzkowski, K., Horodecki, P., Sanpera, A. and Lewenstein, M. (1998). On the volume of the set of separable states. *Phys. Rev. A*, **58**, 883.