Variational Quantum Information Processing

Variational quantum information processing

A DISSERTATION PRESENTED

BY

JHONATHAN ROMERO FONTALVO

TO

THE DEPARTMENT OF CHEMISTRY AND CHEMICAL BIOLOGY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN THE SUBJECT OF

CHEMICAL PHYSICS

HARVARD UNIVERSITY

CAMBRIDGE, MASSACHUSETTS

JANUARY 20TH 2019

Dissertation Advisor: Professor Alán Aspuru-Guzik Jhonathan Romero Fontalvo

Variational quantum information processing

ABSTRACT

Traditional quantum algorithms with guaranteed performance generally require fully coherent quantum computers to operate, making error-correction a necessity. In contrast, variational quantum algorithms seek to remove this requirement by formulating the computation as the approximate optimization of a functional. This optimization is carried out in the space of heuristic functions defined by a parameterized (variational) quantum circuit. Taking advantage of the flexibility in the definition of these heuristics, we can expand the operability of variational algorithms within the regime of coherence times of noisy intermediate-scale quantum devices, avoiding the necessity of error-correction. In this dissertation, we present the application of the variational quantum computing approach to problems in quantum simulation, quantum state preparation, quantum error-correction, and generative modeling.

In the first part, we investigate the use of the variational quantum eigensolver (VQE) for simulating the ground state of fermionic systems. We start by studying the implementation of VQE using the unitary coupled cluster (UCC) ansatz and propose strategies to reduce its cost. We use these insights to carry out some of the first experimental quantum computations of molecular energies using simplified versions of this ansatz. We also propose an extension of UCC to study problems in condensed matter physics, along with a new low-depth circuit ansatz for preparing non-gaussian fermionic states on quantum computers. We show the potential of our approach to describe strongly correlated fermionic systems.

In the second part, we develop three new variational quantum algorithms for problems in quantum computing and machine learning, namely: 1) the quantum autoencoder, to compress ensembles of quantum states, 2) the variational quantum error corrector, to find device-tailored quantum encoding and recovery circuits for quantum error correction, and 3) the variational quantum generator, to generate classical probability distributions. These techniques offer efficient ways to design new quantum circuits for state preparation, to find more effective error-correcting codes and to perform generative modeling with quantum computers. Our work provides insights into the design of variational quantum algorithms and establishes practical guidelines to implement these methods on near-term quantum computers, as well as some future research directions for this field.

Contents

1 INTRODUCTION **1**
1.1 Basic concepts in quantum computing . . . 9
1.2 Variational quantum algorithms . . . 15
1.3 Outline of the main chapters . . . 21

I Progress in variational algorithms for quantum simulation 30

2 VARIATIONAL QUANTUM SIMULATION OF CHEMISTRY USING THE UNITARY COUPLED CLUSTER ANSATZ **32**
2.1 Introduction . . . 32
2.2 Background . . . 35
2.2.1 Quantum chemistry in second quantization . . . 35
2.2.2 Classical ab initio approaches to quantum chemistry . . . 37
2.2.3 Unitary coupled cluster . . . 42
2.3 Variational quantum eigensolver for UCC . . . 43
2.3.1 Implementation of UCC on a quantum computer . . . 44
2.3.2 Choice and preparation of the reference state . . . 48
2.3.3 Energy measurement . . . 50
2.3.4 Parameter optimization . . . 51
2.3.5 Gradient evaluation for UCC . . . 53
2.3.6 VQE-UCC with an active space approximation . . . 58
2.4 Numerical assessment . . . 60
2.4.1 Classical simulation of VQE-UCC . . . 60
2.4.2 VQE-UCC results for H_4 molecular systems . . . 61
2.5 Discussion . . . 74
2.6 Appendix . . . 76
2.6.1 Commutativity of subterms in excitation operators . . . 76
2.6.2 Mapping fermionic operators to qubit operators . . . 78

3 A LOW-DEPTH CIRCUIT ANSATZ FOR VARIATIONAL QUANTUM SIMULATION OF FERMIONIC SYSTEMS **80**
3.1 Introduction . . . 80
3.2 Generalized variational quantum eigensolver . . . 81
3.2.1 Formulation of the problem . . . 82
3.2.2 Bogoliubov unitary coupled cluster theory . . . 83
3.2.3 Generalized Hartree-Fock theory . . . 85
3.2.4 The quantum subroutine . . . 89
3.2.5 Gradient Evaluation for LDCA . . . 99
3.3 Numerical examples . . . 102
3.3.1 Fermi-Hubbard model . . . 103
3.3.2 Cyclobutadiene . . . 105
3.4 Discussion . . . 108
3.5 Conclusion . . . 110

4 TOWARDS COMPUTING MOLECULAR ENERGIES ON NEAR-TERM QUANTUM DEVICES **111**
4.1 Introduction . . . 111
4.2 VQE implementation on near-term quantum devices . . . 114
4.2.1 H_2 in a minimal basis . . . 119
4.2.2 LiH in a minimal basis . . . 124
4.3 Experimental results . . . 129
4.3.1 H_2 simulations . . . 129
4.3.2 LiH simulation on a multiqubit trapped-ion quantum computer . . . 138
4.3.3 Discussion . . . 140
4.4 Conclusions . . . 141

II New variational algorithms for quantum computing and statistical modeling 144

5 QUANTUM AUTOENCODERS FOR EFFICIENT COMPRESSION OF QUANTUM DATA **146**
5.1 Introduction . . . 146
5.2 Quantum Autoencoder Model . . . 148
5.3 Implementation of the quantum autoencoder model . . . 153
5.4 Application to quantum simulation . . . 157
5.5 Discussion . . . 166
5.6 Appendix . . . 169
5.6.1 Fidelity derivation . . . 169

6 A VARIATIONAL ALGORITHM FOR DEVICE-TAILORED QUANTUM ERROR CORRECTION **170**
6.1 Introduction . . . 170
6.2 Background: quantum error correction . . . 173
6.3 Quantum variational error correction algorithm (QVECTOR) . . . 176
6.4 Numerical simulations . . . 179
6.5 Conclusions and outlook . . . 185
6.6 Appendix . . . 188
6.6.1 Noise models . . . 188
6.6.2 Numerical simulation of QVECTOR . . . 191
6.6.3 Accuracy of average fidelity estimate from approximate unitary 2-design . . . 195

7 GENERATIVE ADVERSARIAL QUANTUM MACHINE LEARNING FOR CONTINUOUS DISTRIBUTIONS **199**
7.1 Introduction . . . 199
7.2 Background . . . 202
7.2.1 Classical and quantum generative adversarial learning (GANs) . . . 202
7.3 The variational quantum generator architecture . . . 206
7.3.1 Architecture . . . 206
7.3.2 Training and cost function . . . 211
7.4 Implementation . . . 218
7.4.1 Numerical simulations . . . 218
7.4.2 Implementation on NISQ devices . . . 223
7.5 Conclusion and outlook . . . 226
7.6 Appendix . . . 228
7.6.1 Variational circuits architectures . . . 228
7.6.2 Estimation of analytical gradients for variational circuits . . . 230
7.6.3 Automatic differentiation . . . 231

III Outlook and conclusions **233**

8 FUTURE RESEARCH DIRECTIONS **235**

9 CONCLUSIONS **242**

To Laura, my parents and all my unconditional and loving family.

Acknowledgments

A Ph.D. is definitely not an easy task. Through my own journey, I have been continuously challenged to solve problems and achieve goals that at first sight looked impossible to me. In many occasions, I felt unsecured, but ultimately I was able to overcome my lack of confidence thanks to the continuous support of many people that believed in my abilities and guided me towards many of the answers.

I want to start by thanking my great advisor Alán Aspuru-Guzik, for his commitment to advance science in the most humane and enjoyable way possible. The care he puts on his students and postdocs, his strategic advice, incredible scientific input, and enthusiasm are qualities hard to find in a single scientist. I admire him the most. Many thanks to the other two members of my committee: professors Kang-Kuen Ni and Eric Heller. Professor Ni has been a mentor throughout my doctoral studies, not only as a member of my committee but also during my roles as student and teacher assistant. Professor Heller has been a continuous source of guidance, helping me advance my scientific perspectives through many inspiring conversations. Having scientists with such experience and diverse perspectives in my committee has been a privilege. I also thank my Master's thesis advisor in Colombia, Andres Reyes, who encouraged me to pursue my doctoral studies in first place.

Many of the challenges I faced would have been impossible to overcome without the help of the incredible Aspuru-Guzik group. The content of this dissertation is the product of the many scientific interactions I had with them, some of the most brilliant scientists I have ever met. At the beginning of my Ph.D. studies, I had the fortune to work with Ryan Babbush and Jarrod McClean, who introduced me to the world of quantum computing for Chemistry and for whom I have great respect and admiration. I thank them for all their scientific input and guidance in those first years, that led to collaborations and research projects that made an important part of this dissertation. I also thank Borja Peropadre, Gian Giacomo Guerreschi, and Salvatore Mandrá, postdocs in the quantum subgroup, who helped a then disoriented graduate student to navigate a research group and a research field that were new to him.

Many of the members of the quantum subgroup left around the second year of my Ph.D. to continue their scientific careers as senior scientists in different institutions and companies. But the Aspuru-Guzik group is continuously renovating, and with their farewell, new scientists arrived. Among them, four new postdocs: Jonny Olson, Peter Johnson, Yudong Cao, and Pierre-Luc Dallaire-Demers, with who I embarked in a scientific journey that produced many of the ideas presented in this dissertation. That journey also opened new prospects for our professional careers, ultimately inspiring us to start a new project called Zapata

Computing, a startup on quantum algorithms. What began as a mere contemplation became a reality thanks to the efforts and contributions of many people, among them Alán and the now CEO of Zapata, Christopher Savoie. Today Zapata is a growing and dynamic group of great scientists and professionals advancing the field of quantum computing towards the practical frontier. Among them Jonny, Peter, Yudong, Pierre-Luc, and Borja. I could not be more excited about the prospect of joining them and the rest of the Zapata team in the near future.

I will be always indebted to many of my fellow PhD students, who collaborated with me on many projects and from who I have learned tremendously: Hannah Sim, Ian Kivlichan, Nicolas Sawaya, Benjamin Sanchez, Florian Häse, Tereza Tamayo, Jennifer Wei and Tim Mencke. I also thank the continuous help and encouragement of other members and visitors of the new and old Aspuru-Guzik groups: Semion Saikin, Peter Love, Christoph Kreisbeck, Libor Veis, Loïc Roch, Doran Bennett, Daniel Tabor, Matthias Degroote, Stephanie Valeau, Jacob Sanders, Dmitry Zubarev, among many others.

I also thank Cornelius Hempel, Christine Maier, and Peter O'Malley, with who I had the opportunity to collaborate in the experiments reported in this dissertation. Their patient and commitment to these projects secured their success despite the difficulties we experienced.

I am grateful to all the people and institutions that funded and supported my doctoral studies. Special thanks to the staff of the Chemistry department, for their continuous assistance, and to Marlon Cummings, Siria Serrano, and Felix Negron, the lab administrators of the Aspuru-Guzik group through my Ph.D., whose work behind scenes made possible the everyday functioning of our group.

This Ph.D. also gave me the opportunity to grow in other aspects of my life, assuming other roles that extended my vision of the world and made me reflect on my social obligations as a scientist. Part of it has been my work in Science Clubs, which have given me the opportunity to keep in contact with my country by helping students all over Colombia to fall in love with science. Thanks to the people who offered me the opportunity to be part of this project, Antonio Copete, and Maier Avendaño, and the incredible team of volunteers at Science Clubs Colombia and its recently created spin-off, ScienteLab.

I also thank the Harvard Colombian Student Society, for which I had the honor to serve as treasurer, and my fellow members of the executive board, Santiago Pulido, Juan Pablo Caicedo, and Jose Gonzales. This adventure taught me many lessons about leadership and public service and gave me a new perspective of my country through its small mirror reflection at Harvard.

Finally, I would like to thank who have been through all these struggles and successes my unconditional partner, Laura Castellanos. Her love has been a permanent source of comfort and motivation. For her, I have no words to express my gratitude.

I thank my unconditional and loving family, whose effort has put me where I am today. Thanks to my parents, Carlos Romero and Ana Fontalvo, my sisters Jessica Romero and Liliana Romero, and all my family in Barranquilla. Likewise, I thank who I consider my second parents, Luis Fontalvo and Carmen Gonzales, and my cousins, who are more like siblings, along with the rest of my extended family in Bogota.

In this journey, I also had the pleasure to count with old and new friends, who had been a constant source of encouragement and enjoyment. Many thanks to the ELP group, who had been almost like a family in Boston: Alejandro, Sol, Silvia, Wasay, Boris and Lukas, and to all my Colombian friends who I will not list to avoid the risk of leaving someone out. I love them all.

This has been an amazing learning experience both professionally and personally and I couldn't be more grateful with the Universe for offering me this incredible opportunity.

Citations to previously published work

At the moment of submitting this dissertation, the content of chapters 2 to 7 has appeared as published articles or preprints. I acknowledge the scientific contributions of all the authors of these publications to the present document.

P. O'Malley, R. Babbush, I. Kivlichan, J. Romero, J. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, et al., "Scalable quantum simulation of molecular energies", *Phys. Rev. X*, **6**, 031007 (2016).

J. Romero, J. P. Olson, and A. Aspuru-Guzik, "Quantum autoencoders for efficient compression of quantum data", *Quantum Sci. Technol.* **2**, 045001 (2017).

P. D. Johnson, J. Romero, J. Olson, Y. Cao, and A. Aspuru-Guzik, "QVECTOR: an algorithm for device-tailored quantum error correction", *arXiv preprint arXiv:1711.02249*, (2017).

P.-L. Dallaire-Demers, J. Romero, L. Veis, S. Sim, and A. Aspuru-Guzik, "Lowdepth circuit ansatz for preparing correlated fermionic states on a quantum computer", *arXiv preprint arXiv:1801.01053*, (2018).

J. Romero, R. Babbush, J. R. McClean, C. Hempel, P. J. Love, and A. Aspuru-Guzik, "Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz", *Quantum Sci. Technol.* **4**, 014008 (2018).

C. Hempel, C. Maier, J. Romero, J. McClean, T. Monz, H. Shen, P. Jurcevic, B. P. Lanyon, P. Love, R. Babbush, A. Aspuru-Guzik, R. Blatt, and C. F. Roos, "Quantum chemistry calculations on a trapped-ion quantum simulator", *Phys. Rev. X*, *8*, 031022 (2018).

J. Romero and A. Aspuru-Guzik, "Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions", *arXiv preprint arXiv:1901.00848*, (2018).

Throughout the course of my doctorate, I also contributed to the papers and preprints listed below. Although not part of the present document, these articles either correspond to reviews on topics directly related to this dissertation or incorporate some of its developments.

Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. Sawaya, S. Sim, L. Veis and A. Aspuru-Guzik., "Quantum chemistry in the age of quantum computing", *Submitted*, (2018).

Y. Cao, J. Romero, and A. Aspuru-Guzik, "Potential of quantum computing for drug discovery", *IBM Journal of Research and Development*, ISSN: 0018-8646 (2018).

S. Sim, Y. Cao, J. Romero, P. D. Johnson, and A. Aspuru-Guzik, "A framework for algorithm deployment on cloud-based quantum computers", *arXiv preprint arXiv:1810.10576* (2018).

J. R. McClean, I. D. Kivlichan, D. S. Steiger, Y. Cao, E. S. Fried, C. Gidney, T. Häner, V. Havlíček, Z. Jiang, M. Neeley, et al., "OpenFermion: The electronic structure package for quantum computers", *arXiv preprint arXiv:1710.07629*, (2017).

E. S. Fried, N. P. Sawaya, Y. Cao, I. D. Kivlichan, J. Romero, and A. Aspuru-Guzik, "QTorch: The quantum tensor contraction handler", *PloS one*, **13**, e0208510 (2018).

J. Olson, Y. Cao, J. Romero, P. Johnson, P.-L. Dallaire-Demers, N. Sawaya, P. Narang, I. Kivlichan, M. Wasielewski, and A. Aspuru-Guzik, "Quantum information and computation for chemistry", *arXiv preprint arXiv:1706.05413* (2017).

J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms", *New. J. Phys*, **18**, 023023 (2016).

1

Introduction

Computers are machines designed to store and process information faster and more reliably than humans by themselves could. Nowadays, we employ these machines for a wide range of activities, including secure data storage, communication, prediction, and modeling. Computers are embedded in almost every single human activity, becoming an essential driver for the economy, as evidenced by the emergence and rapid growth of information technology companies[59]. What made this revolution possible was the development of electronic computers, which work thanks to a rather simple but powerful principle in computer science: that *any* possible computation can be broken down into a series of Boolean operations acting on a register of bits[67,178,243,337] (variables taking values 0 or 1). Electronic computers employ electrical circuits to implement bits and Boolean operations, and perform computations by *compiling* sets of instructions written by humans, also called *algorithms*, all the way down to their set of fundamental physical operations [*], which are subsequently executed on hardware[357].

However, the fact that electronic computers can implement *universal* computation does not imply that they can always do this *efficiently*. The word efficiently in this context refers specifically to how fast computational resources (memory and depth [†] required by the sequence of operations composing the computation) grows as the size of the problem increases. It is asserted as a premise in computer science that efficient algorithms employ resources that grow polynomially with the size of the problem, meaning proportionally to a (rather low) power of the problem size. In contrast, exponential scaling is considered inefficient. For some computational problems, the most

[*]Electronic computers implement a *complete* basis[178] of Boolean functions. Any Boolean function can be expressed as a composition of operations in this set.

[†]The depth of a Boolean circuit is the maximum number of gates on any path from the input to the output[178].

efficient algorithms known scale exponentially with system size. One example of such problems is factoring, for which the most efficient known algorithm scales proportionally to $exp(cn^{1/3})$, with n being the number of digits (in binary) of the number we want to factor[260].

Another example of a problem with an exponential scaling is the simulation of the dynamics of arbitrary quantum systems. Consider for example a system with n spins. Each spin is represented with a binary basis, corresponding to basis states $|0\rangle$ and $|1\rangle$, such that a system of n spins requires 2^n basis computational states, $|x_1, x_2, \ldots, x_n\rangle$ with $x_i \in \{0, 1\}$. Correspondingly, the state of this system is described by a vector of 2^n complex amplitudes, $c_{x_1,x_2,\ldots,x_n}$, whose squared norm is the likelihood of finding the system in the corresponding computational state, $|x_1, x_2, \ldots, x_n\rangle$. While certain classes of quantum states admit a more efficient representation of the state vector (e.g., stabilizer[122] or matrix product[250] states), this is not always the case. Consequently, just storing arbitrary quantum states becomes inefficient on a classical computer [‡]. The problem gets worse if we consider studying the *dynamics* of quantum states. For long time intervals, this requires multiplication by unitary matrices of size $2^n \times 2^n$. For this reason, it is believed that simulating the dynamics of an arbitrary quantum system on classical computers is, in general, inefficient[178].

Conversely, the hardness associated with quantum dynamics can constitute a resource. This brings us to the question of whether or not quantum systems themselves can be used as computers. The field of quantum computing grew out of this inquiry, originally addressed by the early works of Manin[217], Feynman[108] and Deutsch[88,89], among others. Their research answered this question about quantum computing positively and derived a series of abstract models that formulate quantum computation independently of a particular physical implementation. Among those models, the most widely used and the basis for writing quantum programs nowadays is the quantum circuit model. Quantum circuits comprise a set of qubits (two-level quantum systems, with basis $|0\rangle$ and $|1\rangle$, just like spins) and quantum unitary operations, known as quantum gates. A quantum circuit is a series of quantum gates acting on a set (or *register*) of qubits, where each operation acts on a subset of the qubits.

Provided the correct set of quantum gates, a quantum circuit can implement *universal*

[‡]The number of classical resources needed to represent a given quantum state is related to the degree of *entanglement* of the state. States with low entanglement admit low-rank representations and therefore require less classical resources.

computation, meaning it can implement any possible Boolean function[178,309]. The basis of the procedure goes as follows. Suppose we wish to implement a function: $F : \mathbb{B}^n \to \mathbb{B}^m$, where $n \leq m$, without loss of generality. For that purpose, we implement a quantum computation using a quantum circuit U, such that:

$$U|x_1,\ldots,x_n,0,\ldots,0\rangle = \sum_{y_1,\ldots,y_m} c_{y_1,\ldots,y_m}|y_1,\ldots,y_m\rangle \tag{1.1}$$

Where we have assumed that we can initialize the qubits in a known initial state, for example, a computational state $|x_1,x_2,\ldots,x_n\rangle$. This initial state encodes the information regarding the input for the function, and correspondingly, the states $|y_1,\cdots,y_m\rangle$ encode potential outputs. However, unlike classical computers, the output of the quantum computer is a quantum state with an exponential number of amplitudes. To read the answers from the quantum register, we need to perform measurements on the output quantum state. A particular binary string, $y_1,\ldots,y_m$ is measured with probability $|c_{y_1,\ldots,y_m}|^2$, as pointed out earlier. Therefore, to implement F we need to design a sequence of quantum gates (a *quantum algorithm*) such that the resulting circuit, U, prepares an output state that produces $F(x_1,\ldots,x_n)$ with probability close to 1 when measured. Described this way, we see that a quantum model of computation is possible. However, does it offer any advantage compared to classical computing?

It turns out there are problems for which it is possible to formulate a quantum algorithm that is more efficient than its classical counterpart. An example of such problems is quantum simulation. Consider, for example, a system of quantum spins as mentioned earlier. The quantum states of such a system can be represented using only a linear number of qubits (one qubit per spin) in contrast to the exponential number of bits needed to represent a general state of the system. Furthermore, the evolution under a unitary operator e^{-iHt}, where H is a qubit Hamiltonian, can be efficiently implemented on a quantum computer when H is a sum of local interactions[210]. In recent years, the prospect of simulating complex quantum systems using quantum computers has motivated the development of sophisticated techniques for quantum simulation[117] with application in many research fields, including chemistry[62].

The other two prominent problems where quantum algorithms show advantage are unstructured database search and integer factorization. In the first case, a quantum algorithm proposed by Grover[128] provides a quadratic speed-up compared to classical approaches. In the second case, an algorithm proposed by Shor[308] is capable of factoring an integer of bit-size n using approximately n^3 operations, an exponential speed-up compared to the best known classical algorithm, mentioned earlier. Grover's and Shor's algorithms had a profound impact on the theoretical quantum information community, constituting the blueprint for many quantum algorithms designed to exhibit a quantum speed-up[236]. Shor's algorithm especially had a significant impact within the field of quantum computing, due to its potential for breaking the RSA protocol, widely used for secure data transmission[276]. The disruptive potential of Shor's method brought significant attention from academia, industry, and government, leading to a substantial increase of the resources devoted to quantum computing research.

The impact of Shor's algorithm and the development of techniques to control quantum systems[136,364], set the stage for the first physical realizations of quantum computers in the 2000's[41,91,198,247,258,365]. To be considered a quantum computer, a particular physical implementation must meet the requirements established by DiVicenzo[92], which are linked to the ability to implement and execute quantum circuits accurately. The endeavor of building quantum computers has advanced substantially in the last decade, delivering different types of scalable physical implementation, also called quantum computing architectures or platforms. Nowadays, a substantial part of the development of these platforms is carried out at big research facilities by major technological companies. Their current focus is on building a quantum computing device capable of executing a task that cannot be simulated using existing traditional computing resources. This milestone has been coined *quantum supremacy*[268]. Some of the problems proposed for quantum supremacy experiments are specially-crafted theoretical tasks without a practical commercial application[43,241]. In this situation, one might ask the question: when will quantum computers be able to execute useful algorithms? To answer this, we first need to know how what quantum computational resources are required to implement a given practical algorithm, such as Shor's factoring, and then determine how long it will take for the emerging quantum computing

industry to develop those capabilities. It turns out that for an algorithm such as Shor's, current estimates indicate that practical instances might require on the order of hundreds of millions of physical qubits to factor a two thousand digit number[111]. At the moment of writing this document, the biggest quantum processors reported does not comprise more than a hundred qubits[129].

It might come as a surprise that implementing Shor's algorithm requires such a large amount of resources. The reason for this lies in the almost counter-intuitive logic behind building a quantum computer. On the one hand, qubits need to be isolated from their environment to prevent decoherence [§] to corrupt the information encoded in quantum states. On the other hand, we need to be able to address qubits through external physical operations to implement gates and measurements. As a result, decoherence is always present at a certain level, causing physical implementations of unitary operations to have a finite error rate, ε. Correspondingly, errors accumulate along the execution of the quantum circuit driving the probability of error close to 1 after only $L \sim \varepsilon^{-1}$ operations[178]. Despite this, large scale quantum computing is still possible in principle thanks to the development of *quantum error-correction codes* and the theory of fault-tolerant quantum computation. These protocols encode the information of a single *logical* qubit using several physical qubits and provide procedures to implement logical gates as well as to detect and correct physical errors. Effective implementations of quantum error correction require the error rate to be below a certain error level or *threshold*, which is determined by the specific type of quantum-error correcting code and physical implementation.

The price to pay for error-correction is a significant overhead in qubit number and run time, as illustrated by the example of Shor's algorithm mentioned earlier. Consequently, it is believed that achieving scalable error correction, although viable, is a long-term goal still decades away. However, while industry and academia work to accomplish this goal, noisy digital quantum computers with an increasing number of resources will become available. This period of quantum technologies has been recently coined as the noisy intermediate-scale quantum (NISQ) era[269]. NISQ devices will not have the resources to implement error correction and consequently will not be capable of executing large instances of practical algorithms such as Shor's. However, these

[§]Decoherence is the irreversible degradation of a quantum state caused by interactions with its environment. One could describe this process, for example, as a superoperator acting on the pure state of a quantum register.

machines will be big enough to implement quantum dynamics that cannot be simulated with existing classical computing resources, allowing the first demonstrations of quantum supremacy[43]. Given that current proposals for quantum supremacy do not have practical applications, one might wonder if *practical* quantum supremacy is possible using NISQ devices.

Some tentative answers to the question of practical quantum supremacy, or *quantum advantage*, have emerged through the development of new quantum algorithms which take into account the limitations of NISQ devices. These approaches are the quantum variational eigensolver (VQE) for quantum simulation and the quantum approximate optimization algorithm (QAOA) for solving combinatorial optimization problems. QAOA, developed by Farhi et al.[104], was initially conceived as a method to approximately solve constraint satisfaction problems (CSP) using quantum circuits with low-depth. The algorithm starts by mapping the cost function of the CSP to a diagonal Hamiltonian on qubits. Solutions to the CSP problem are encoded as the state prepared by a quantum circuit parameterized by two variables. Correspondingly, by minimizing the expectation value of the Hamiltonian for these parameters, one gets an approximate solution to the problem. Similarly, the VQE algorithm, initially proposed by Peruzzo and McClean et al.[255], aims at finding approximations to the ground state of a Hamiltonian using a parameterized quantum circuit. Here, this circuit plays the role of a *variational ansatz*, such that approximate solutions are guaranteed by the variational principle of quantum mechanics[126]. By computing the variational state and its observables on the quantum computer, the original formulation of VQE offers efficient approximate solutions to the time-independent Schrödinger equation.

The VQE and QAOA algorithms have paved the way for a new paradigm in quantum algorithm design. Previous to these proposals, quantum algorithmic development mostly focused on designing algorithms with a provable advantage over classical ones, such as Shor's. These algorithms were designed to be executed entirely on a fully coherent quantum computer for which error-correction is a necessity. In contrast, VQE and QAOA, seek to remove this requirement by formulating the computation as the approximate optimization of a functional in the space of heuristic functions defined by a parameterized (variational) quantum circuit. The flexibility in the definition of these heuristics allows to expands the operability of these approaches within the

regime of the coherence times of noisy intermediate-scale quantum devices, avoiding the necessity of error-correction. The strategies employed by both algorithms to achieve such robustness and flexibility are similar, and so we present them together in the form of three basic concepts:

1. *Approximate quantum computing*: Both, QAOA and VQE offer *approximate* solutions to their corresponding computational problems, as opposed to an exact solution guaranteed within certain precision. In other words, we remove performance guarantees on the algorithm in exchange for more flexibility in the number of resources required by the algorithm to operate. The concept of approximate quantum computing captures the possibility of finding a practical advantage for a given computational task even without those performance guarantees. This advantage might come, for example, as a practical reduction in the time required for solving a computational task, or as achieving a better accuracy for a given instance of the problem compared to a classical approximation.

2. *Variational quantum circuits*: Variational quantum circuits (also called parameterized quantum circuits) are quantum circuits where some of the unitary operations are tuned as part of the algorithm execution. The tuning can be performed through a particular parameterization of the quantum gates. Computational problems encoding solutions as extreme values of a functional can be naturally described using variational circuits as heuristic functions. The flexibility in the selection of the circuit offers a lot of operational robustness, allowing to choose circuits that can operate even within the limits of the coherence times of NISQ devices, thus removing the requirement of error correction.

3. *Hybrid quantum-classical computing*: QAOA and VQE require an optimization routine to tune the parameters of the variational circuit. This optimization outer-loop is implemented using a classical computer interfacing with the quantum computer via classical communication. This feature motivates the concept of hybrid quantum-classical (HQC) computing. In the HQC framework, we subdivide a computational task into subtasks, some of which are executed by a quantum computer while the rest are solved classically. By assigning to the quantum computer tasks for which it can offer an advantage, it might be

possible to create algorithms that exhibit advantage using fewer resources than the corresponding algorithm that implements the entire original task on a quantum device.

The principles described above constitute a new approach to computing using quantum devices that we designate as *variational quantum information processing*. Correspondingly, we call the algorithms developed using this approach *variational quantum algorithms*. This dissertation covers original research on this topic. We divide our exposition in two parts: 1) progress in variational algorithms for quantum simulation and 2) new variational algorithms for quantum computing and statistical modeling.

The first part of this dissertation began with the VQE algorithm as a starting point. Our goal was to improve the performance of VQE applied to problems in Chemistry and to study its implementation on NISQ devices. To that aim, we introduced established techniques in traditional quantum chemistry to improve the performance of the VQE algorithm when implemented with a Unitary Coupled Cluster ansatz[279]. Along the same line of work, we designed some of the first proof-of-principle scalable demonstrations of the VQE algorithm for molecules and implemented them on superconducting qubits[248] and trapped ions[142] architectures. Simultaneously, we explored the application of the UCC ansatz to problems in condensed matter physics, such as superconductors. To that aim, we develop a variant of UCC that employs a general fermionic gaussian state as reference state in the calculation. We proposed a method for preparing such gaussian fermionic states with circuit depths that scale only linearly in the size of the system. Using this circuit as the starting template, we developed a new VQE ansatz for studying strongly-correlated fermionic systems[78].

The research in the second part of this dissertation started as an exploration of the connections and similarities between variational algorithms and techniques employed in machine learning and statistical modeling[39,120]. As a result of this cross-pollination, we developed three new variational algorithms. The first one was the quantum autoencoder (QAE) algorithm[280], inspired by the homonymous method in classical machine learning. QAE employs a quantum variational approach to compress sets of quantum states, offering a practical way of learning new circuits for state preparation, a task common to all quantum algorithms. QAE also provides a pragmatic view of the

concept of quantum compression in quantum information theory. The parallel between classical and quantum autoencoders kept inspiring new algorithm developments. In particular, the use of classical autoencoders as de-noising models prompted us to explore the method of QAE for mitigating errors on a quantum computer. This inception gave rise to a new variational algorithm for designing encoding and recovery circuits for quantum error correction, called QVECTOR (quantum variational error corrector)[163]. The main feature of this approach is that the generated encoding and recovery circuits are tailored to the noise of the specific device.

Finally, inspired by recent advances in classical generative modeling[119], we explore the use of variational quantum circuits to model classical probability distributions, concluding in the development of the variational quantum generator (VQG). This approach combines a variational circuit with a method to encode a classical random variable into a quantum state to build a generative model[278]. VQG integrates several classical machine learning techniques such as adversarial learning[121] and establishes a practical way of combining variational circuits with neural networks, ultimately contributing to the nascent field of quantum machine learning[38].

Throughout the rest of this dissertation, we will present in detail the concepts and research results introduced above. We will start by describing the necessary background, which includes a brief introduction to quantum computing basics (Section 1.1), followed by a general overview of variational algorithms, and an outline of the main chapters (Section 1.2). Then, we will proceed to Chapters 2 to 7, which correspond to the detailed research papers that compose this work. Each of these chapters is mostly self-contained regarding its notation and technical background. Finally, Chapter 8 will present a short perspective on the development of variational quantum algorithms, emphasizing applications to quantum chemistry and machine learning. We will close our exposition with some final remarks.

1.1 Basic concepts in quantum computing

This section is a brief introduction to the basic concepts of quantum computing and the language necessary to navigate this dissertation. For an in-depth introduction to the topics of quantum information and quantum computing we refer the reader to Ref.[178,243].

QUANTUM INFORMATION

Information in quantum computers is encoded in quantum states. Instead of composing with the direct product, as with classical states, quantum states are composed using the tensor product rule. For example, in the quantum circuit model, general states are expressed as tensor products of two-level quantum systems or *qubits*. The state of a single qubit can be conveniently represented using a Bloch sphere (See Figure 1.1(a)). Correspondingly, an arbitrary state of a qubit can be expressed as a superposition of two orthogonal basis states. By convention, the computational basis corresponds to $|0\rangle = \binom{1}{0}$ and $|1\rangle = \binom{0}{1}$, such that an arbitrary qubit state is written as $\alpha|0\rangle + \beta|1\rangle$, where α and β are normalized complex amplitudes. Correspondingly, a state vector in n qubits is described by a unit vector in $\mathbb{C}^{2^n}$.

Another aspect of the quantum representation of information is measurement. Extracting information from a quantum state requires measurement, whose outcome statistics are governed by Born's rule. To measure a quantum state we project it to a specific basis. One example is measurement in the computational basis comprised of all the possible tensor products of states $|0\rangle$ and $|1\rangle$ in n qubits. The conventional one-qubit basis in quantum computing is given by the eigenstates of Pauli matrices:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}; \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}; \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{1.2}$$

where the eigenstates of the Z matrix correspond to the computational basis. An n qubit state in a one-qubit basis can be written as $|\psi\rangle = \sum_{x_1,\ldots,x_n} c_{x_1,\ldots,x_n} |x_1,\ldots,x_n\rangle$ such that the state $|x_1,\ldots,x_n\rangle$ is observed with probability $|c_{x_1,\ldots,x_n}|^2$ when $|\psi\rangle$ is measured in this basis. Correspondingly, quantum states can be interpreted as probability vectors with an ℓ^2-norm normalization condition, compared to the ℓ^1-norm normalization condition of the corresponding discrete probability vector.

The theory of quantum information studies the use of quantum states to represent information, establishing a connection to classical information theory[243]. Correspondingly, some of the concepts in classical information theory such as entropy have their counterparts in quantum theory.

Property	Classical	Quantum
Combining systems	$S_1 \times S_2$	$\|S_1\rangle \otimes \|S_2\rangle$
Normalization	$\sum_k p_k = 1$	$\mathrm{Tr}[\rho] = 1$
Positivity	$p_k \in \mathbb{R}^+$	$\langle k\|\rho\|k\rangle \geq 0$
Expectation values	$\sum_k p_k f(k)$	$\mathrm{Tr}[\rho F]$ where $F = F^\dagger$
Marginals	$\sum_k p_{km} = p_m$	$\mathrm{Tr}_B[\rho^{AB}]$
Entropy	$\sum_k p_k \log p_k$	$\mathrm{Tr}[\rho \log \rho]$

Table 1.1: Comparison of properties of quantum and classical information. The normalization and positivity properties are associated to a probabilistic interpretation of quantum states. We also compare the formulas for expectation values, marginals, and entropy of classical probability distribution and quantum states. For a diagonal density matrix, the classical and quantum formulas for entropy become equivalent. (*Adapted from Ref.[358]*).

In quantum information theory, it is sometimes convenient to represent quantum states as density operators. For a pure state, its density matrix is defined as $\rho = |\psi\rangle \langle\psi|$. The density matrix formalism enables the description of *quantum ensembles*, which correspond to classical probability distributions over a set of quantum states, $\rho = \sum_k |\psi_k\rangle \langle\psi_k|$. Table 1.1 contrasts some of the basic elements and properties of classical and quantum information in the density matrix formalism.

Another important aspect of quantum information is the property of entanglement. An entangled quantum state of a joint system is a state that cannot be factored as a tensor product of states of the individual components, independently of the basis used to represent the total system. An example of an entangled state in two qubits is $|00\rangle + |11\rangle / \sqrt{2}$. One way to quantify entanglement is through the Schmidt rank, which can be calculated using Singular Value Decomposition given a specific partitioning of the system[243]. Perturbative approaches or techniques based on matrix product states can better approximate quantum states with relatively low entanglement but struggle with highly entangled (strongly correlated) states[64]. This difficulty motivates the idea that quantum computers can offer an advantage simulating strongly correlated systems, such as those arising naturally in many problems of condensed matter physics and quantum chemistry. Entanglement is one of the ingredients necessary for quantum computation to provide a computational advantage[166].

Quantum circuit model

There exist several abstract models of quantum computing, including quantum Turing machines[88], adiabatic quantum computing[105], quantum cellular automaton[218], among others. However, the

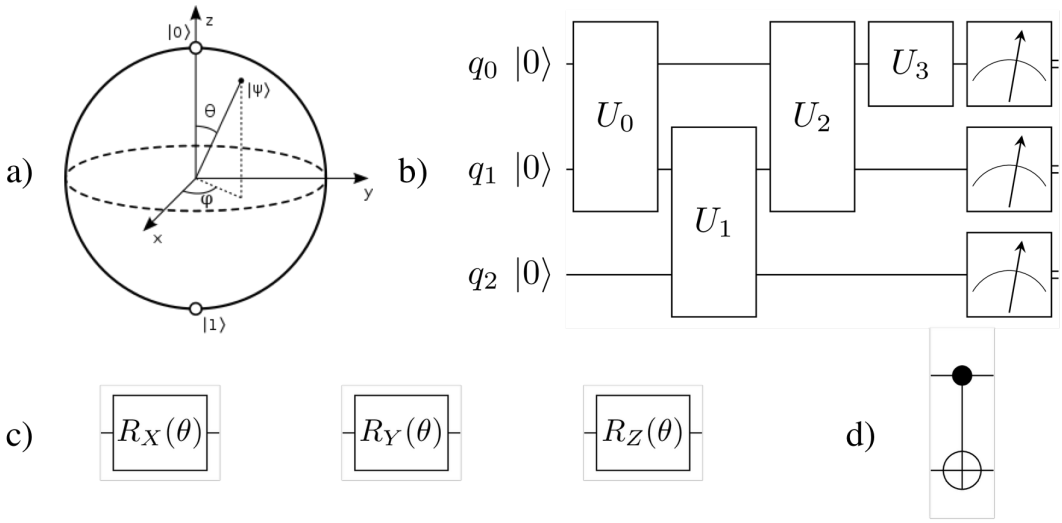


Figure 1.1: Some basic elements of quantum computing: a) Bloch sphere representation of a qubit: the state of a qubit can be described as $|\psi\rangle = cos(\theta/2)\,|0\rangle + e^{i\phi} sin(\theta/2)$. X, Y and Z rotations can be pictured as rotations in the corresponding axes. b) Example of a quantum circuit: the circuit acts on three qubits, represented by the wires labeled as q_i, and comprises 3 two-qubits gates and 1 single-qubit gate, represented by the rectangles. The 'meter' symbol represents a measurement in the Z basis. c) Graphical representation of single qubit rotations. d) Graphical representation of a CNOT gate.

most popular and the one we employ throughout this dissertation is the quantum circuit model.

As outlined above, a quantum circuit starts with a register of n qubits. The register is generally initialized in some fiducial state, usually $|0\rangle^{\otimes n}$. A quantum algorithm is specified by a sequence of unitary matrices acting on specific sets of qubits, $U_i[S_i]$, where S_i represent the set of qubits indexes. Correspondingly, the unitary matrix U_i has size $2^{|S_i|} \times 2^{|S_i|}$, where $|S_i|$ is the number of qubits in the set S_i. These unitary operations are also called quantum gates. The output state prepared by the quantum circuit corresponds to:

$$U_d[S_d]U_{d-1}[S_{d-1}]\ldots U_0[S_0]\,|0\rangle^{\otimes n} \tag{1.3}$$

where the indexes of the unitary operations indicate the time order in which operations are applied. We can write quantum circuits using a graphical representation similar to that of classical Boolean circuits, as illustrated in Figure 1.1(b). Due to the unitary character of quantum gates, quantum circuits are reversible, meaning that no information is lost during the circuit execution. Correspondingly, it can be shown that quantum circuits with arbitrary unitary operations

implement reversible classical Boolean functions, which in turn can perform any classical Boolean function[178,243]. The demonstration of this result is what regards the quantum circuit model as a universal model of computation. Furthermore, in analogy with the concept of complete basis in classical Boolean logic, there are finite *universal gate sets*. Using only the operations within the set, it is possible to approximate an arbitrary unitary operation in n qubits. Some of the universal gate sets include gates with continuous parameters, for instance, the universal set comprising arbitrary single qubit rotations and CNOT gates (defined below). Other universal sets contain discrete operations, just as in classical Boolean logic. Universal discrete gate sets play an essential role in the implementation of error-correction protocols.

The selection of a particular gate set depends on the types of physical interactions realizable in a particular quantum computing architecture. Nonetheless, most of the modern architectures based on a quantum circuit model implement arbitrary single qubit rotations and CNOTs, or another two-qubit operation capable of composing a CNOT when combined with single-qubit gates. These gates have the following definition:

$$R_M(\theta) = exp(-i\theta M/2) \quad M \in \{X,Y,Z\}; \quad \text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{1.4}$$

where θ is the rotation angle. 1.1(c)-(d) illustrate the graphical circuit representation of these gates. In addition to a limited set of gates, physical realizations of quantum computers also have constraints in the sets of qubits on which gates can be effected. Those constraints correspond to the device *connectivity*. For example, a fully connected array of qubits can implement two-qubit gates at the hardware level between all possible qubit pairs. In contrast, a linear connectivity allows to implement two-qubit gates only on adjacent pairs of qubits for a given ordering of the qubits. Then, an algorithm written in the abstract circuit model, needs to be compiled to the native set of the architecture taking into account its connectivity, which has an impact on the final depth of the algorithm.

PHYSICAL REALIZATIONS OF QUANTUM COMPUTERS AND ERROR CORRECTION

As mentioned above, a physical realization of a quantum computer needs to have the capabilities described by DiVicenzo's criteria, listed below:

1. Scalable: the physical system implements well-characterized qubits and the cost of adding more qubits or implementing more gates scales efficiently regarding physical resources.

2. Initializable: the ability to initialize the state of the qubits on a simple fiducial state, for example, $|0\rangle^n$.

3. Coherent: sufficiently long coherence times, such that we can implement algorithms with success probabilities close to 1.

4. Universal: it should implement a universal set of quantum gates, meaning a set of unitary operations that can generate any arbitrary unitary operations on the qubits.

5. Measurable: a measurement capability on individual qubits.

Quantum computing platforms meeting this criteria include superconducting qubits[356], trapped ions[134,235], solid-state qubits[156], photonic systems[349], among others. Currently, superconducting qubits and trapped ion platforms are ahead regarding the number of quantum resources offered[285]. We present a more detailed description of the physics of these devices is Section 4.2.0.1. Some of these early quantum computers are also available to the public via cloud-based services[233].

As mentioned earlier, the physical implementation of quantum computers is inevitably noisy. The theory of error correction and fault-tolerant quantum computing provides a series of protocols to ameliorate the effects of imperfect preparation, encoding, processing, decoding and readout of the computer. The central insight of these protocols is to represent the information of a single physical qubit using multiqubit states, representing *logical* qubits. The quantum circuits implementing the mapping from the space of physical qubits to logical ones are called encoders. Information encoded in logical qubits can be protected errors in the quantum computer again by implementing *recovering* operations: series of quantum operations that detect and correct physical

errors. These two elements are the basis of error-correcting codes. So far, most of these codes have been designed by decomposing noise processes into a discrete set of single-qubit errors. This strategy has made possible the theoretical design of error-correcting codes. Along these, the theory of fault-tolerant quantum computing has been developed to address the problem of implementing fault-tolerant gates on logical qubits. For each specific choice of error-correcting code and fault-tolerant protocol, there exists an error threshold required for the protocol to succeed. In order to implement error-correction, a quantum computing platform needs to have enough physical qubits and operations with errors below the error threshold. A more detailed introduction to the principles of quantum error-correction is presented in Section 6.2. Traditional algorithms, such as Shor's, require fully coherent quantum circuits to properly function and therefore their practical use is conditioned on the availability of error-corrected quantum computers. Variational algorithms, on the other hand, try to get around the need of error-correction by introducing a new paradigm for quantum algorithm design.

1.2 Variational quantum algorithms

Variational quantum algorithms are designed following the principles of variational quantum information processing described above, which include 1) the approximate character of the calculations, 2) the use of variational circuits and 3) the use of a hybrid-quantum classical computing strategy. These three elements integrate a general approach that can be applied to find approximate solutions to problems where the goal is to find a function that extremizes the value of quantities that depend upon this function. For example, in VQE, the goal is to find a quantum circuit that prepares a state that minimizes the energy of a Hamiltonian. In QAOA, the goal is to find the quantum circuit that prepares the state that maximizes the cost function encoding the constraints. Formally, this type of optimization problem is addressed by the calculus of variations[115], a field of mathematical analysis concerned with finding derivatives of functionals [¶].

From this perspective, variational quantum computing processing concerns a strategy for finding extremes of *quantum functionals*, which we define as mappings from a space of quantum states to

[¶] A functional is a mapping that takes a function as the input and returns a scalar as output.

a scalar. The problem specifies the specific form of this functional. For example, in VQE, the functional corresponds to the expectation value of the problem Hamiltonian. If the Hamiltonian can be expressed as a sum of a polynomial number of tensor products of Pauli matrices, the expectation value can be estimated efficiently on the quantum computer[226]. In general, viable quantum functionals need to be efficiently computable using quantum resources. A functional implemented on a classical computer is not feasible because it requires exponential resources to read the output quantum state.

The variational circuit provides the space of functions for the optimization. Recall that quantum circuits implement Boolean functions, and correspondingly, by parameterizing a sequence of quantum gates we generate a set of functions defined by these parameters. Correspondingly, we could formulate the optimization of a quantum functional, C, as follows:

$$\underset{\theta_i}{\operatorname{argmin}} C[F_{\theta_i}(x)]; \quad F_{\theta i} : |z\rangle \xrightarrow{U(\theta_i)} |x\rangle\,, \tag{1.5}$$

where we write the optimization as a minimization without loss of generality and $\{F_{\theta_i}(z)\}$ represents the set of functions to optimize over, where each function is indexed by its parameters θ_i. The set of variational circuits defined by $\{U(\theta_i)\}$ acting on an initial state $|z\rangle$ implement the functions. The parameters θ_i are classical, for example, angles of single-qubit rotations. Since the value of the functional is also classical, we can employ a classical algorithm to carry out the optimization, completing the general formulation of a variational algorithm. A second possibility would be to perform the optimization using quantum resources. To the best of our knowledge, this possibility has not been studied in the literature and constitutes an open research question not addressed in this work.

In practice, variational quantum algorithms are carried out following an iterative procedure comprised of the steps listed below and depicted in Figure 1.2:

1. Preparation of input states: we start by preparing the input state $|z\rangle$. Certain functionals might require evaluating the function on several input states.

2. Application of the parameterized unitary: we apply the parameterized unitary to the input

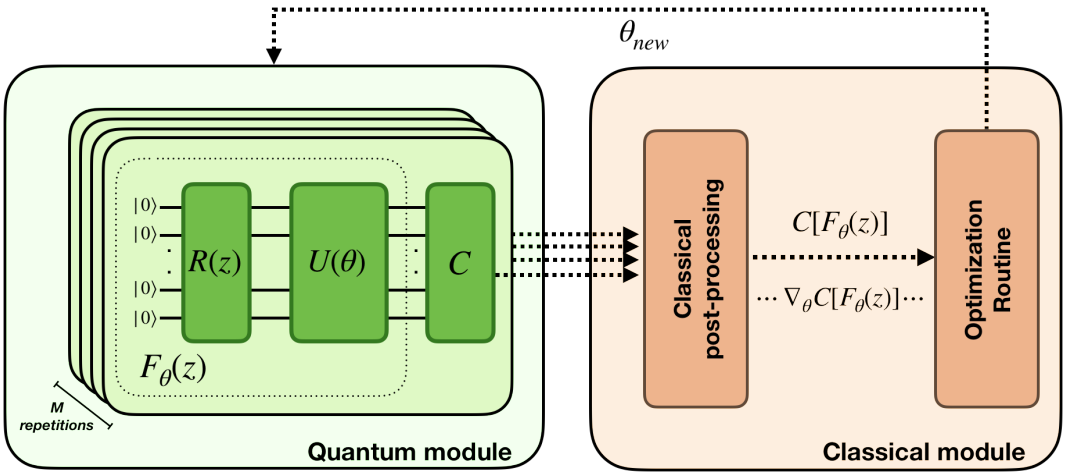


Figure 1.2: Implementation of a quantum variational algorithm using a hybrid quantum-classical computer: the function $F_\theta(z)$ is implemented on the quantum computer by preparing the initial state $|z\rangle = R(z)|0\ldots0\rangle$ and executing the parameterized unitary $U(\theta)$. The circuit C implements the functional computation. To estimate $C[F_\theta(z)]$, the state preparation and necessary measurements are repeated several times to accumulate sufficient statistics. The results are collected and post-processed on a classical computer. Post-processing might include techniques such as statistical modeling. Also, circuit C could be replaced by a circuit to compute derivatives of the functional, which are required by some optimization routines. An iteration of the optimization algorithm generates a new candidate for the optimal variational parameters θ_{new}. The whole cycle between the quantum and classical modules is repeated until convergence. Here, discontinuous arrows represent classical information.

state on the quantum device, preparing the output state $U(\theta_{(n)})|z\rangle$. The parameterized unitary is defined by choice of variational circuit *ansatz*, which should correspond to a quantum circuit that cannot be efficiently computed using classical resources [‖];

3. Functional estimation: The value of the functional is estimated using an efficient approach implemented by a quantum circuit C. Ultimately, the functional estimation reduces to performing measurements on the output state. To estimate the functional up to some accuracy, we need to repeat the preparation of the output state several times to collect sufficient measurement statistics;

4. Classical feedback: based on the previous estimations of the functional, a classical optimization technique proposes a new candidate for the optimal variational parameters θ_{n+1};

[‖]Certain sets of quantum gates compose quantum circuits that allow for efficient classical simulation. Examples are *Clifford* gates (See Chapter 5) and *matchgates* on circuits with linear connectivity (See Chapter 2).

5. Steps 1-4 are repeated until convergence criteria (e.g. ΔC, $\Delta\theta$) are satisfied.

The quantum variational approach has proven incredibly practical regarding its implementation, establishing these algorithms as candidates for the first demonstrations of practical quantum computing applications[269]. In particular, significant research efforts are currently devoted to the development of VQE[62,368] and QAOA[46,106,130,211,373]. The similarities of quantum variational approaches with variational inference and neural network models have also caused significant growth in the number of quantum machine learning approaches employing variational circuits[298]. The relation between variational quantum circuits and neural networks is central to many of the ideas proposed in this work, and therefore we describe it in more detail in the subsection below. Other ideas developed for quantum computing with NISQ devices have also improved the prospects of applying variational algorithms to practical problems in the near-term. In particular, the development of error mitigation techniques[98,169,331] will have a positive impact on the accuracy of variational algorithms. These methods are different from error-correction in the sense that they do not provide fault-tolerant quantum computing. Instead, their main goal is to reduce the impact of errors in implementations at the physical level, without encoding quantum information into logical qubits. It is important to point out as well that despite being motivated by the limitations of NISQ devices, variational algorithms can be implemented using error-correction techniques.

Given that variational algorithms are heuristic and provide approximate solutions, one might wonder whether there will be value in these approaches once error-correction becomes available, especially compared to quantum algorithms with a proven advantage on the same tasks. The first thing to consider is that designing algorithms with proven advantage is challenging and so far only a relatively small number of such algorithms have been discovered[236]. Therefore, for many applications, it is possible that variational algorithms will be the only quantum algorithms available in the error-correction era. A second consideration is that variational algorithms are designed to optimize the use of quantum resources, and therefore it is likely that their error-corrected implementations are more efficient than implementations of their counterparts with proven advantage. Consequently, quantum algorithms might be still competitive with traditional algorithms regarding computational cost. Furthermore, since formal demonstrations of advantage

are based on asymptotic considerations, there is a chance for variational algorithms to be more efficient and sufficiently accurate for specific instances of a problem. Finally, the third consideration is that variational algorithms can be combined with traditional quantum algorithms to obtain even more powerful approaches. A concrete example of this can is VQE. Techniques employed for simulating quantum systems with provable speed-up, such as quantum phase estimation[62,117,178], require the preparation of states with sufficient overlap with the eigenstates of the Hamiltonian to measure eigenvalues with high probability. Using VQE, it will be possible to prepare such states, ultimately boosting the success probability of phase estimation.

Connection between variational quantum circuits and neural networks

Artificial neural networks are parameterized mathematical models that can be trained to mimic high dimensional real functions. The simplest type of artificial neural network is the feed-forward type, which consists of a series of non-linear transformations of an input vector $x \in \mathbb{R}^n$. The product of each transformation corresponds to a *layer*. Non-linear transformations are composed by applying a non-linear function element-wise to a linear transformation of x. More specifically, a single layer of a neural network is implemented by a mapping of the form $F_0^1 : \mathbb{R}^{d_0} \to \mathbb{R}^{d_1}$ defined as:

$$F_0^1(x) = f(W_0^1 x + b_0), \tag{1.6}$$

where the matrix $W_0^1 \in \mathbb{R}^{d_0 \times d_1}$ and the vector $b \in \mathbb{R}^{d_1}$ act as parameters known as weights and biases, respectively. The function f is the non-linear scalar function also called *activation* function. Popular choices of activation functions include sigmoid and tanh, among many others[39]. The output of applying one transformation can be fed as input in a second transformation, such that layers with variable dimensions can be concatenated to compose very complex functions. For example, a neural network with n layers and the same activation acting on each transformation has

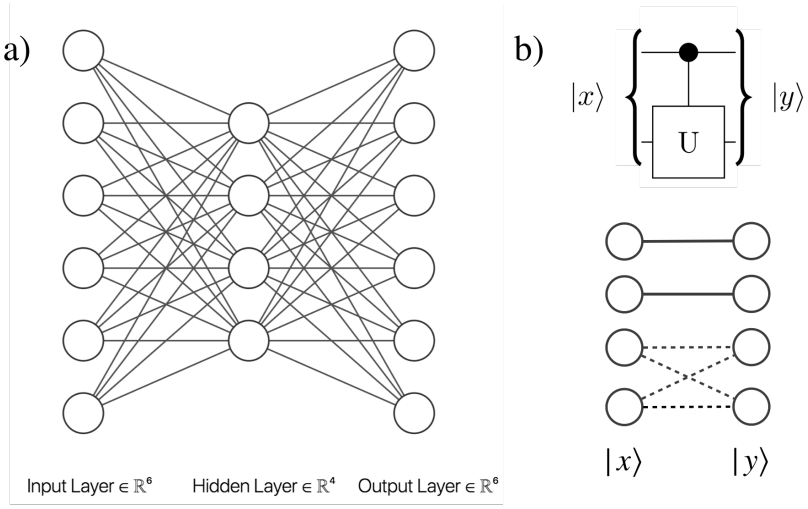


Figure 1.3: Connection between neural networks and variational quantum circuits: a) Graphical representation of a feed-forward neural network with three layers of sizes six, four and six respectively. After the input layer, each new layer is the result of applying a linear transformation followed by the application of a non-linear activation function. Internal layers are designated as *hidden* layers. b) Illustration of the connection between variational circuits and neural networks. The controlled gate, shown as an example above implements a linear unitary transformation indicated by the neural network graph below. Each layer represents a two-qubit state vector in $\mathbb{C}^4$. We only show weights with norm different from zero. Weights with norm 1 appear as solid edges while those with norm below one appear as discontinuous edges.

the form:

$$F_0^n(x) = f\left(W_{n-1}^n\left(\dots f(W_1^2 f(W_0^1 x + b_0) + b_1)\dots\right) + b_{n-1}\right) \tag{1.7}$$

This intricate composition can be more easily pictured as a graph, as illustrated in Figure 1.3(a). Each layer is represented as a set of nodes in the graph. The edges connecting the nodes correspond to the parameters in the weight matrix. This graphical representation immediately resembles the quantum circuit model. Furthermore, variational quantum circuits correspond to neural networks with unitary weight matrices acting on complex vectors of size 2^n, where n is the number of qubits, as shown in Ref.[296]. This connection is illustrated in Figure 1.3(b) for the case of a two-qubit gate acting upon a two-qubit state. Notice that a variational quantum circuit on n qubits implements a matrix of complex weights of dimensions $2^n \times 2^n$, but, unlike typical neural networks, not all the entries of this matrix act as parameters. The linear transformations implemented by variational circuits are parameterized by only $O(poly(n))$ variables,

corresponding to the tunable parameters of the gates. In this sense, variational quantum circuits can be interpreted as efficient parameterization of linear transformations in high dimensions.

1.3 Outline of the main chapters

The research presented in this dissertation advances the field of variational quantum algorithms in two directions:

1. **Part I: Variational algorithms for quantum simulation**: we developed improved techniques for applying and implementing the VQE algorithm to compute the ground state energies of molecules and systems in condensed matter physics. In the same direction, we designed a new variational circuit for simulating fermionic systems. The depth of this new ansatz scales only linearly with the size of the system.

2. **Part II: Variational algorithms for quantum computing and statistical modeling**: we introduced three new variational quantum algorithms with application to the problems of quantum compression, quantum error-correction, and classical generative modeling, respectively.

The four algorithms studied here follow the blueprint for variational quantum algorithms outlined in the previous section. Table 1.2 describes and contrasts these approaches according to their application, initial state(s) employed, the general layout of the variational circuit used and functional. In what follows we offer a summary of the content of the main chapters and its significance.

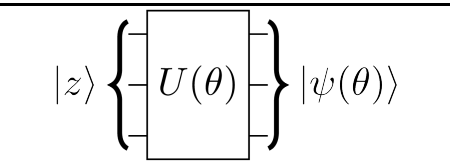
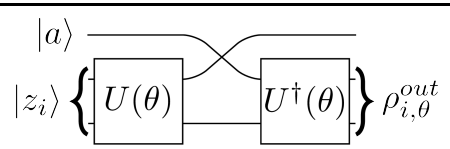
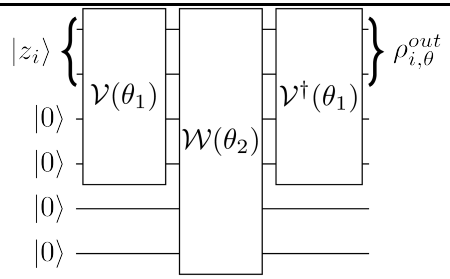
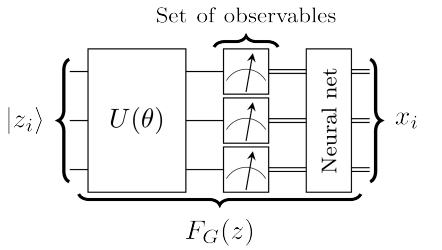
Algorithm	Application	Initial state(s)	Variational circuit layout	Functional	Chapter(s)
VQE	Quantum simulation	Computational basis state corresponding to mean-field approximation		Expected value of the Hamiltonian, $\langle\psi(\theta)\|H\|\psi(\theta)\rangle$, $H=\sum_i H_i$	2-4
QAE*	Compression of quantum states; state preparation	Ensemble of states $\sum_i p_i \|z_i\rangle$ to compress. Reference state $\|a\rangle$.		Average fidelity, $\sum_i p_i \cdot F(\|z_i\rangle, \rho_{i,\theta}^{out})$	5
QVECTOR*	Finding device-tailored encoding ($\mathcal{V}$) and recovering ($\mathcal{W}$) circuits for quantum error-correction	A set of states approximately sampled from the Haar-measure using an approximate 2-design circuit, S_i, $S_i\|0\cdots0\rangle = \|z_i\rangle$		Average fidelity, $\frac{1}{N}\sum_i^N F(\|z_i\rangle, \rho_{i,\theta}^{out})$, where $\theta=[\theta_1,\theta_2]$	6
VQG*	Modeling classical probability distributions	States generated by encoding a random variable z using a parameterized quantum circuit, $R(z_i)\|0\cdots0\rangle = \|z_i\rangle$		Probability of fooling a discriminator F_D, $-\frac{1}{N}\sum_i^N[\log F_D(x_i)]$. F_D and F_G are trained with adversarial learning.	7

Table 1.2: Summary of the quantum algorithms presented in this dissertation: We compare VQE, QAE, QVECTOR and VQG in terms of the description of variational circuits presented in Section 1.2, which comprises the functions $F_{\theta_i}(z)$, implemented by the applying the variational quantum circuits $U(\theta_i)$, to an initial state $|z\rangle$. Output states appear as $|x\rangle$ or as ρ_θ^{out}. The circuit layout illustrates the disposition of the variational circuits during training; circuit wires do not necessarily correspond to a single qubit. Notice that the VQG model is hybrid, incorporating a classical neural network, and its final output, x, is classical. VQG is trained using an adversarial learning approach, where a second model F_D learns to discriminate samples generated by VQG from those coming from the probability distribution to be modeled (See Chapter 7 for more details). F_D can be either a classical or a quantum model. Algorithms marked with * are new algorithms developed as part of this dissertation.

PART I: PROGRESS IN VARIATIONAL ALGORITHMS FOR QUANTUM SIMULATION

This part presents developments in the variational quantum eigensolver (VQE) algorithm applied to problems in quantum chemistry and condensed matter physics. Table 1.2 describes the essential elements of VQE, where a trial state is prepared by applying a parameterized unitary to a reference state generally obtained using a mean-field method. The mean-field calculation provides the basis for the second quantized representation of the system Hamiltonian. This Hamiltonian can be mapped to a qubit Hamiltonian using a classical efficient mapping. Mean-field states can be generally prepared using shallow depth quantum circuits.

VQE was proposed initially in Ref.[255], as an alternative to phase estimation to compute approximate ground state energies of molecules on near-term quantum devices. This early work included a proof-of-principle non-scalable implementation [**] of VQE using a two-qubit quantum photonic chip. As part of this proposal, the Unitary Coupled Cluster ansatz, a classically intractable model Chemistry, was postulated as a candidate ansatz for VQE[369]. While it was known that UCC could be implemented using techniques similar to those employed for phase estimation[359], this approach had not been tested.

Chapter 2 summarizes the results of this assessment. Here, we studied the application of VQE to the simulation of molecular energies using the unitary coupled cluster (UCC) ansatz. We introduced new strategies to reduce the circuit depth for the implementation of UCC and improved the optimization of the wavefunction using efficient classical methods to approximate the optimal variational parameters. Additionally, we proposed an analytical approach to compute the energy gradient. Our proposal reduces the sampling cost for gradient estimation by several orders of magnitude compared to numerical gradients. To illustrate these improvements, we performed numerical simulations for a system of four hydrogen atoms that exhibit strong correlation and showed that we could reduce the depth of the UCC circuit ansatz without introducing significant loss of accuracy in the final wavefunctions and energies.

[**] In the context of quantum computing, scalable implementations refer to implementations where the required classical and quantum resources increase in the same way that they would for an arbitrarily large problem. In this case, the experiment was not scalable because the Hamiltonian matrix was loaded into a classical computer and diagonalized as a pre-processing step.

Another realization of the UCC study was that despite heuristic improvements, the formal scaling of this ansatz was significant for near-term devices, as also pointed out in other works[137]. About the same time, quantum circuits with depths that scale at most linear in the size of the register became the new goal in algorithm development. This situation prompted us to search for improvements to the UCC ansatz that could reduce quantum resources and also expand its application to problems in other fields apart from quantum chemistry. We started by looking at the limitations of the original UCC formulation to describe systems in condensed matter physics, such as superconductors, where the system Hamiltonian does not commute with the number operator. Relaxing this requirement in mean-field theory leads to the generalized Hartree-Fock method[338], for which the mean-field solution is a general fermionic Gaussian state[187]. We reformulated UCC with this new reference state as presented in Chapter 3, where we show how to prepare fermionic gaussian states using a linear-depth circuit of quantum matchgates. By augmenting the set of available gates with nearest-neighbor phase coupling (ZZ interactions), we engineered a low-depth circuit ansatz (LDCA) that can accurately prepare the ground state of correlated fermionic systems. We illustrated our approach with numerical simulations of LDCA applied to small instances of the Hubbard model and to an extended Hubbard model Hamiltonian that mimics cyclobutadiene. Our new variational circuit ansatz extended the range of applicability of VQE to systems with strong pairing correlations such as superconductors, atomic nuclei, and topological materials.

As quantum devices became available, it was evident that progress in both the algorithmic and the hardware side was going to require the realization of proof-of-principle implementations of quantum algorithms. The modest resource requirements and flexibility of VQE compared to other algorithms, positioned this approach as one of the first milestones in the pipeline of quantum computing experiments. Based partially in the results of Chapter 2, we designed proof-of-principle implementations of VQE that were carried out in collaboration with some of the leading groups in superconducting qubits and trapped ions devices. In the first demonstration, we implemented VQE to calculate the potential energy surface of the ground state of molecular hydrogen in a minimal basis set, using an array of superconducting qubits. From these results, we predicted the correct dissociation energy to within chemical accuracy of the numerically exact result. In the second

demonstration, we implemented VQE to calculate the potential energy surface of the ground states of molecular hydrogen and lithium hydride on a multi-qubit trapped ion device. We compared different encoding methods using up to four qubits and analyzed the impact of varying error sources in the results. Our analysis provided guidelines for mitigating errors and ways to circumvent potential roadblocks in the execution of VQE on NISQ devices.

PART II: NEW VARIATIONAL ALGORITHMS FOR QUANTUM COMPUTING AND STATISTICAL MODELING

The general formulation of variational quantum computing offers a blueprint for designing variational quantum algorithms to fit different purposes. The connection between variational circuits and neural networks provided us with a source of inspiration for models that admitted a formulation in the language of quantum variational algorithms. The first of such models was the autoencoder approach employed in machine learning. Classical autoencoders are neural networks that can learn efficient low dimensional representations of data in higher dimensional space. The task of an autoencoder, given a set of inputs $\{x_i\} \in \mathbb{R}^{d_1}$, is to map each of this points to a lower dimensional point $\{y_i\} \in \mathbb{R}^{d_2}$, such that x_i can likely be recovered from y_i. This transformation can be modeled using a neural network with intermediate layers of size d_2 and input and output layers with size d_1. By training the model to reproduce the value the input at the end of the network, it is possible to obtain the desired transformation. Inspired by this idea, we introduced the model of a quantum autoencoder (QAE) to perform similar tasks on quantum data. Table 1.2 summarizes the basic elements QAE, where a quantum ensemble of states is compressed using a unitary operation implemented with a variational circuit. To optimize the circuit parameters, we sampled from the ensemble and performed a cycle of compression and decompression, such that the functional corresponds to the average fidelity of reconstructing the input states. In our original proposal, this fidelity is estimated using a technique called SWAP test[52]. In further developments of the algorithms, we employed a simpler circuit inversion method to estimate such fidelity[311]. We applied the QAE algorithm in the context of quantum simulation to compress ground states of the Hubbard model and molecular Hamiltonians. The encoding circuits learned with QAE can be used

to design new circuits for state preparation, which in this particular case, allows generating new variational circuits for VQE.

Another use of classical autoencoders is as de-noising models. In this case, noise is added to the input layer, such that the network learns to remove noise from noisy data. Typically, this application requires a neural network comprising hidden layers with more dimensions than the input layer. The strategy of mapping information to a higher dimensional space to build a denoising map is also the basis of error-correction. This realization inspired us to explore the application of the variational quantum approach to the problem of error correction. Specifically, we designed a variational algorithm where variational circuits are trained as encoding and recovery circuits by optimizing the average fidelity of performing these operations on a quantum device. Therefore, the error-correcting protocols obtained with this approach are expected to optimally address the *actual* noise in the machine, as opposed to that of an artificial or approximate noise model. The essential elements of this algorithm, called variational error corrector (QVECTOR), are summarized in Table 1.2. QVECTOR is presented in Chapter 6, where we develop this approach for the task of preserving quantum memory and analyzed its performance with numerical simulations. Our results showed that subject to phase damping noise; the simulated QVECTOR algorithm can learn a three-qubit encoding and recovery which extend the effective T_2 of a quantum memory six-fold. Subject to a continuous-time amplitude- plus phase-damping noise model on five qubits, the simulated QVECTOR algorithm learned encoding and decoding circuits which exploit the coherence among Pauli errors in the noise model to outperform schemes that do not leverage such coherence. As described earlier, current proposals for fault-tolerant quantum computation require a significant overhead on quantum resources and will not enable useful quantum computation on near-term devices of 50 to 100 qubits. With QVECTOR we aimed at reducing this overhead, providing an approach that can be implemented in existing quantum hardware and on a myriad of quantum computing platforms.

Finally, in Chapter 7, we explored the use of variational quantum circuits in a classical task: modeling classical probability distributions. To do this, we developed the quantum variational generator (VQG) approach. The architecture of this model incorporates two elements: a quantum

circuit employed to encode a classical random variable into a quantum state, called the *quantum encoder*, and a variational circuit whose parameters are optimized to mimic a target probability distribution. Samples are generated by measuring the expectation values of a set of pre-defined operators. Our *quantum generator* can be complemented with a classical function, such as a neural network, as part of the classical post-processing, as described in Table 1.2. To train the VQG model we employed an adversarial learning approach, where the quantum generator learns through its interaction with a discriminator model that compares the generated samples with those coming from the real data distribution. We showed that our quantum generator could learn target probability distributions using either a classical neural network or a variational circuit as the discriminator. In developing VQG, we introduced software tools and techniques from traditional machine learning, such as automatic differentiation, to perform the optimization of the variational circuits. The strategies developed for VQG can be applied to other problems in machine learning, serving as a blueprint for designing quantum machine learning models for NISQ devices.

SIGNIFICANCE OF THIS WORK

The introduction of VQE and QAOA in 2014 coincided with the transition of experimental quantum computing from mostly an academic and public endeavor to an industrial enterprise. In the last five years, we have witnessed a rapid development of quantum computing architectures towards the goal of commercializing quantum computing services[233]. Many businesses now look at quantum computing as a potential source of disruption, which has attracted more public and private investment in quantum technologies. As a result, a quantum computing ecosystem incorporating hardware and algorithm developers as well as potential users of the first NISQ devices has emerged. Many of these potential users expect quantum computers to provide answers to difficult practical problems; stakes on quantum computing are at its highest. The inevitable question is: will these early imperfect quantum computers deliver such answers? We hope this is the case, and we believe that a viable path towards a definite affirmative answer lies in the further development and perfecting of quantum variational approaches.

This dissertation collects some of the early works on variational algorithms, which provides

examples of the type of challenges we will face to push variational approaches to the practical frontier, along with some strategies to tackle these challenges. Specifically, the work of part I highlights the importance of bringing specific domain knowledge into the design of the algorithms. While the variational quantum framework is very general, its practical application to a field requires adaptation of the algorithm to the problem. A lot of the specific domain knowledge can be incorporated through the design of the variational circuit. Examples of this are the application of efficient classical approximations to improve the efficiency and performance of VQE with a UCC ansatz in Chapter 2 and to simplify the experimental implementation of this algorithm in Chapter 4. An example of a variational circuit tailored to a particular application is the low-depth circuit ansatz of Chapter 3. Currently, a general strategy for designing variational algorithms to general problems does not exist. A good starting point to start building this strategy is to develop metrics to assess the "quality" and "representation power" of several variational circuits. Some fellow scientists have begun to work in this direction.

Optimization is another critical aspect of variational algorithms. The field of numerical optimization is large[246], and therefore identifying and adapting existing optimization techniques that are suited to quantum variational approaches requires significant effort. Another part of this challenge is finding efficient quantum schemes to evaluate the corresponding functional and their derivatives. Our work made progress in this direction by proposing techniques to evaluate gradients of variational circuits in Chapter 2, and adopting optimization strategies and software tools from classical machine learning, as illustrated by the use of adversarial learning techniques and automatic differentiation in Chapter 7. In this sense, exploiting the parallels between neural networks and variational circuits can also help to build a better theoretical understanding of the representation power of quantum circuits and the difficulties associated with the learning process.

Underlying the comparison between the representation power of neural networks and quantum circuits is the permanent question of quantum advantage. This question haunts Chapter 7, where we develop the variational quantum generator approach. This method combines a classical neural network and a quantum circuit to perform a task that could be accomplished exclusively with a classical neural network. Can the hybrid quantum-classical approach provide an advantage? For

specific generative tasks, variational circuits have been shown to offer a formal advantage using complexity theory arguments[94]. The difficulty of simulating certain probability distributions generated by quantum circuits is also the argument behind quantum supremacy experiments[43]. How these results apply to the VQG model is still an open question. It is also an open question whether any proven formal difference bare a *practical* advantage compare to what classical computers can do today. Nonetheless, the formulation of hybrid quantum-classical machine learning models such as VQG not only allow us to formulate these questions, but also provide the practical means to start testing these type of models on real quantum devices. If a model like VQG proves useful for research and commercial applications, the range of applications of NISQ devices will expand tremendously.

Finally, we point out that the success of variational approaches ultimately depends on the quality of the quantum devices and the quantum operations they implement. Therefore, maximizing the utility of quantum devices, in particular, early machines, is another way to push quantum computing towards the practical frontier. These improvements can be introduced at the software level by finding more efficient protocols to implement quantum operations common to many algorithms. Tasks within this category are what the algorithms presented in Chapters 5 (QAE), and 6 (QVECTOR) can accomplish. These methods are autonomous and data-driven approaches to design more efficient protocols for quantum state preparation and quantum error-correction, and as such, they work as tools that can serve other quantum algorithms to improve their performance and reduce their resource requirements. Just as happened with classical computers, quantum computers might help design themselves. QAE and QVECTOR enable these machines to do so.

The work presented here is only a part of the continuously growing body of research in variational algorithms[129]. The history of this field has just started, and many exciting developments await for us.

Part I

Progress in variational algorithms for quantum simulation

This page intentionally left blank.

2

Variational quantum simulation of chemistry using the unitary coupled cluster ansatz

Apart from minor modifications, this chapter originally appeared as[279]:

"Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz". Jonathan Romero, Ryan Babbush, Jarrod R. McClean, Cornelius Hempel, Peter J. Love and Alán Aspuru-Guzik. *Quantum Sci. Technol.*, **4**, 014008 (2018). Copyright (2018) by IOP Publishing.

2.1 Introduction

The solution to the time-independent Schrödinger equation for molecular systems allows for the prediction of chemical properties, holding the key to materials discovery and catalyst design[76,101,133,154]. Despite advances in the field of quantum chemistry, many relevant problems such as the prediction of chemical rates and the description of transition-metal complexes remain challenging[29,219]. These difficulties stem from the approximate nature of classically tractable quantum chemistry approaches, which often fail in the description of strongly correlated systems[214,328]. In addition, the application of exact methods, such as exact diagonalization of the electronic Hamiltonian, require exponential resources with current classical algorithms, limiting the exact simulation of molecular energies to systems comprising only a few atoms[140,141].

Feynman envisioned that quantum computers could provide a tractable way to simulate quantum systems[108]. This idea, formalized by Abrams and Lloyd a decade later[4], has been developed into a series of quantum algorithms for quantum simulation[117,171,370]. The first algorithm extending these

approaches to the calculation of molecular energies was proposed by Aspuru-Guzik et al.[14]. This first proposal, further developed in[359], combines Trotterization of the molecular Hamiltonian and phase estimation (PEA) to compute the ground state energy of a molecule.

Early studies on the quantum resources required by this algorithm showed that the circuit depth scales as $O(N^8)$[137], where N the total number of spin-orbital functions. Fortunately, numerical studies indicated that the scaling for real molecules is closer to $O(N^6)$[265] or $O(Z_{\text{max}}^3 N^4)$ when trying to simulate ground states. Here, Z_{max} is the largest nuclear charge of the molecule[18]. Recent proposals have developed new algorithms for this problem by considering simulation based on Taylor series methods as opposed to Trotterization[16,179], performing simulations in a fixed particle number manifold[17,47,324,334], and considering specialized basis functions[19,179]. Despite these recent theoretical improvements, all phase estimation based algorithms for this problem are unlikely to solve classically intractable molecules without error-correction. The variational quantum eigensolver (VQE)[226,255,353] is a an alternative algorithm that is closer to near-team applicability due to lower coherence time requirements.

The VQE algorithm finds the best variational approximation to the ground state of a given Hamiltonian for a particular choice of ansatz. This task is achieved by two subroutines. The first subroutine employs a quantum computer to prepare a parameterized wavefunction ansatz and measure the expectation value of the Hamiltonian given a set of values for the parameters. The second subroutine consists of an optimization algorithm running on a classical computer. The optimization algorithm employs the quantum subroutine as an objective function and finds the parameters that minimize the energy of the ansatz. This procedure offers several advantages that make it a candidate for exploiting the performance of near-future quantum devices: adaptability to different quantum architectures, intrinsic robustness to quantum errors[224,248] and a smaller coherence time requirements[226].

The VQE approach was first applied to the simulation of molecular energies. In this case, a trial wavefunction is prepared by the application of a parametrized unitary, followed by the calculation of the energy via Hamiltonian averaging[222,226]. The value of the energy is minimized using a classical optimization routine that updates the variational parameters. Accordingly, the final cost of the calculation depends on the number of iterations required for convergence and the amount of operations involved in each preparation and measurement cycle of the quantum subroutine. This optimization scheme has been experimentally demonstrated in different quantum platforms, including photonic chips[255], ion traps[142,304] and superconducting circuits[168,248].

Traditionally, a unitary coupled cluster (UCC) approach has been used as the ansatz for the state preparation[226,255,369]. This method provides a hierarchy of wavefunctions that can be prepared on a quantum computer using a polynomial number of gates and it is believed to provide better accuracy than classical coupled cluster[25,72,103,150,189], which is generally regarded as the “gold standard" of quantum chemistry[26]. Despite these advantages, recent studies have pointed out that the number of parameters in UCC might be still too large to allow practical calculations for large molecules[353].

In this chapter, we aim to describe in more detail the implementation of VQE approaches for molecular systems using a UCC ansatz and introduce strategies to improve its efficiency. In Section 2.2, we describe the approaches commonly used in classical quantum chemistry calculations and introduce the UCC ansatz in this context. In Section 2.3, we discuss in detail the implementation of VQE with a UCC ansatz, including the generation of initial guesses and the reduction of computational resources using pre-screening of the cluster amplitudes and active space approaches. In addition, we introduce a method to compute the gradient of the energy with respect to the variational parameters that can be combined with gradient-based optimization methods. In Section 2.4, we illustrate the proposed strategies through numerical simulations of the VQE approach for a variety of chemical systems. Finally, in Section 2.5 we present a brief

discussion of the results.

2.2 Background

2.2.1 Quantum chemistry in second quantization

Within the Born-Oppenheimer approximation, a molecule is comprised of a system of η electrons interacting in the potential produced by nuclei located at fixed positions. We may describe this problem using the formalism of second quantization, where N single-particle spin orbitals can be either empty or occupied. Any interaction between electrons can be represented using annihilation and creation operations, a_p and $a_p^\dagger$, that obey the following anti-commutation relations, associated with fermionic statistics:

$$[a_j, a_k]_+ = 0 \quad [a_j^\dagger, a_k^\dagger]_+ = 0 \quad [a_j, a_k^\dagger]_+ = \delta_{jk} \tag{2.1}$$

where $[a,b]_+ \equiv ab + ba$. In the absence of external fields the non-relativistic molecular Hamiltonian can be written as:

$$H = h_{nuc} + \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s \tag{2.2}$$

where h_{nuc} corresponds to the classical electrostatic repulsion between nuclei, and the constants h_{pq} and h_{pqrs} correspond to the one- and two-electron integrals. Using atomic units, where the electron mass m_e, the electron charge e, Bohr radius a_0, Coulomb's constant and $\hbar$ are unity, we may write:

$$h_{pq} = \int d\sigma \varphi_p^*(\sigma) \left(-\frac{\nabla_{\vec{r}}^2}{2} - \sum_i \frac{Z_i}{|\vec{R}_i - \vec{r}|} \right) \varphi_q(\sigma) \tag{2.3}$$

$$h_{pqrs} = \int d\sigma_1 \, d\sigma_2 \, \frac{\varphi_p^*(\sigma_1)\varphi_q^*(\sigma_2)\varphi_s(\sigma_1)\varphi_r(\sigma_2)}{|\vec{r}_1 - \vec{r}_2|} \tag{2.4}$$

$$h_{nuc} = \frac{1}{2}\sum_{i \neq j} \frac{Z_i Z_j}{|\vec{R}_i - \vec{R}_j|} \tag{2.5}$$

Here Z_i represents the nuclear charge, $\vec{r}$ and $\vec{R}$ denote electronic and nuclear spatial coordinates, respectively, and σ is now a spatial and spin coordinate with $\sigma_i = (\vec{r}_i; s_i)$. Summations run over all nuclei. The function $\varphi(\sigma)$ represent one-electron functions (spin-orbitals) that are often obtained from a mean field calculation such as Hartree-Fock (HF).

After removing the translational and rotational degrees of freedom, the electronic energy of a molecular system is a function of $3q-6$ parameters ($3q-5$ for linear molecules) that we will denote by $\vec{\mathbf{R}}$, where q is the number of atoms. The function $E(\vec{\mathbf{R}})$ is called the potential energy surface (PES). The accurate calculation of the PES is one of the main challenges of quantum chemistry as it is required for predicting and understanding a wide range of chemical processes, such as reaction dynamics, bond-breaking and chemical kinetics.

The prediction of thermochemical properties such as reaction rates determines the accuracy required from ab initio calculations of the PES[257]. Chemical rates, for instance, are exponentially sensitive to changes in the Gibbs free energy, and thus changes in the PES. This sensitivity can be seen from the Erying equation for chemical rates,

$$\text{rate} \propto \frac{e^{-\beta \Delta G^{\ddagger}}}{\beta}, \tag{2.6}$$

where $\Delta G^{\ddagger}$ is the difference in free energy between reactants and transition state and β is the inverse temperature in atomic units. At room temperature and atmospheric pressure, an error ε in $\Delta G^{\ddagger}$ of 1.4 kcal/mol translates to a chemical rate error of a factor of ten. This leads to the definition

of *chemical accuracy* which sets ε to the order of 1 kcal/mol or approximately 1.59×10^{-3} Hartrees (43.3 meV)[141].

2.2.2 Classical ab initio approaches to quantum chemistry

The inherent difficulty of solving the Schrodinger equation for many-electron systems has motivated the development of a series of standard models for the construction and calculation of approximate electronic wavefunctions in quantum chemistry. The simplest approach is to represent the wavefunction as a single anti-symmetrized product of one-electron functions, known as a Slater determinant. The Hartree-Fock method provides such a single-determinant solution. In this scheme, the molecular orbitals are expressed as a linear combination of atomic orbital functions. The combination coefficients are then optimized by a self-consistent variational procedure in which each particle is made to interact with the average density of the other particles. The output of this calculation provides a mean-field approximation to the molecular wavefunction. Unfortunately, the Hartree-Fock method is incapable of approximating the electron correlation effects that are essential for computing energies within or close to chemical accuracy[141].

To correct for this problem, one can expand the wavefunction as a superposition of all the determinants in the η-electron Fock space. The coefficients in the expansion can be parametrized in different ways, defining different models for the description of electron correlation. Two popular parametrizations are the configuration interaction (CI) and the coupled-cluster (CC) methods.

In the full configuration interaction (FCI) approach, which is exact within a given basis, the wavefunction is expanded as a linear combination of all the determinants in the η-Fock space. The coefficients of the expansion can be solved for by variational minimization of the energy, providing the exact wavefunction for a given orbital basis. Unfortunately, the FCI wavefunction becomes rapidly intractable due to the factorial dependence on the number of determinants N related to the total number of spin orbitals[141].

To generate classically-tractable CI approaches one can truncate the CI expansion to include only determinants with a fixed number of excitations with respect to a reference configuration. The reference is usually chosen to be the Hartree-Fock state. This idea can be formalized by defining excitation operators as follows:

$$T = \sum_{i=1}^{\eta} T_i \tag{2.7}$$

$$T_1 = \sum_{\substack{i\in\text{occ} \\ a\in\text{virt}}} t_a^i a_a^\dagger a_i \tag{2.8}$$

$$T_2 = \sum_{\substack{i>j\in\text{occ} \\ a>b\in\text{virt}}} t_{ab}^{ij} a_a^\dagger a_b^\dagger a_i a_j \tag{2.9}$$

where the *occ* and *virt* spaces are defined as the occupied and unoccupied sites in the reference state. In this construction, the operator T_1 generates single excitations from the reference, T_2 generates double excitations and the definition of higher order excitations follows naturally. t_a^i and t_{ab}^{ij} correspond to expansion coefficients. The exact full CI wavefunction is thus,

$$\begin{aligned} |\text{FCI}\rangle &= (1+T)\,|\text{HF}\rangle \\ E_{\text{FCI}} &= \min_{\vec{t}} \frac{\langle \text{FCI}|\, H\, |\text{FCI}\rangle}{\langle \text{FCI}\rangle\, \text{FCI}} \end{aligned} \tag{2.10}$$

where $|\text{HF}\rangle$ is the reference state (for instance, the Hartree-Fock solution) and $\vec{t}$ is the vector comprising the expansion coefficients. The maximum number of excitations allowed, defines the order of truncation, k. The FCI solution can be systematically approached by increasing k. The

computational cost of truncated single-reference CI approaches scales as $O(\eta^k(N-\eta)^{k+2})$, assuming $N, \eta >> k$, as discussed in Ref.[306]. Tractable classical CI truncation is generally limited to single and double excitation operators, which define the CI singles and doubles method (CISD).

The truncated CI expansion suffers from two major problems. First, the method converges slowly when applied to highly correlated systems. To circumvent this problem we can use an entangled reference state that captures the main computational states contributing to the total wavefunction. This is the base of *multireference* methods in quantum chemistry[141,328], which are generally more involved than truncated single reference CI approaches.

The second complication is that configuration interaction is not size-extensive. A method that is size-extensive for a system of non-interacting fragments has a wavefunction that is multiplicatively seperable and an energy that is proportional to the size of the system[141]. This means that the total wavefunction factorizes as a product of the wavefunctions of the independent fragments and the corresponding energy is the sum of the energies of the fragments. These conditions assure that the energy scales linearly with the size of the system. Size-extensivity is a desirable feature for approximate methods in quantum chemistry because many chemical properties, such as the atomization energy, are obtained by subtracting the energy of systems with different sizes. In addition, we expect that higher order expansions must be used for larger molecules if the method is not size-extensive.

The lack of size-extensivity of the truncated CI wavefunction can be overcome by recasting the linear FCI parametrization in the form of a product wavefunction. This is done in the CC method by means of an exponential ansatz:

$$|\Psi\rangle = e^T |\mathrm{HF}\rangle \tag{2.11}$$

where the operator T is defined as for CI. Notice that in this scheme the parameters $\vec{t}$ constitute

excitation amplitudes instead of expansion coefficients. As with CI, CC is usually truncated at some fixed level of excitation. For instance, the method known as coupled cluster singles and doubles (CCSD) is based on the ansatz,

$$|\text{CCSD}\rangle = e^{T_1+T_2}\,|\text{HF}\rangle\,. \tag{2.12}$$

Whereas truncated CI wavefunctions contain contributions from a polynomial number of determinants at a given truncation level, truncated CC wavefunctions have support on all the determinants in the η-Fock space. Tractable implementations of the coupled-cluster theory rely on projecting the Schrödinger equation in the form

$$e^{-T}He^{T}\,|\text{HF}\rangle = E_{\text{CC}}\,|\text{HF}\rangle \tag{2.13}$$

against a set of configurations $\{\langle\mu|\}$. This set spans the space of all the states that can be reached by applying the truncated cluster operator T linearly to the reference state[26]. This treatment generates the following set of non-linear equations for the CC energy and amplitudes:

$$\langle\text{HF}|\,e^{-T}He^{T}\,|\text{HF}\rangle = E \tag{2.14}$$

$$\langle\mu|\,e^{-T}He^{T}\,|\text{HF}\rangle = 0 \tag{2.15}$$

The key point in establishing the size-extensivity of CC theory is to note that the operator $e^{-T}He^{T}$, known as the similarity-transformed Hamiltonian, is additively separable and produces additively separable energies. Similarly, it can be shown that the operator e^{T} is multiplicatively separable and thus generates multiplicatively separable wavefunctions[141].

In practice, the similarity-transformed Hamiltonian is expanded using the

Baker-Campbell-Hausdorff (BCH) formula:

$$\begin{aligned} e^{-T}He^{T} =& H + [H,T] + \frac{1}{2}[[H,T],T] \\ &+ \frac{1}{3!}[[[H,T]T],T] + \frac{1}{4!}[[[[H,T]T],T],T]. \end{aligned} \tag{2.16}$$

The expansion terminates at fourth order due to the commutation properties of excitation operators for the special case that the reference is a single determinant[26,141]. This fact allows for an efficient evaluation of the projected CC equations without further approximation.

While truncated CC is classically tractable and more accurate than truncated CI, there are two substantial weaknesses to the theory. The first weakness is the BCH expansion of the similarity-transformed Hamiltonian is only convergent under the assumption of a single reference state. Consequently, single reference coupled cluster generally performs poorly for strongly correlated systems. This means that coupled cluster is fairly reliable when computing energies at equilibrium configurations but likely to fail for transition states or near dissociation limits of multiple bonds. At those geometries, excited surfaces may become nearly degenerate with the ground state and a single determinant (e.g. the Hartree-Fock state) may have very small overlap with the ground state. Although the field of multireference coupled cluster methods has expanded in the last years, current approaches are still far from being practical for large molecular systems[214].

The second weakness of the projected coupled-cluster formulation is that the operator e^{T} is not unitary and therefore the energy obtained from Eq. (2.14) is not variational. In the next section we discuss a formulation of coupled cluster theory that is variational and can be made multireference. While this formulation is not classically tractable, it can be implemented using a quantum computer.

2.2.3 UNITARY COUPLED CLUSTER

The shortcomings of the traditional coupled cluster ansatz described in the previous section can be overcome by redefining the excitation operator to be unitary, an approach known as unitary coupled cluster (UCC)[25,150,189]:

$$|\Psi\rangle = e^{T-T^\dagger}|\mathrm{HF}\rangle\,. \tag{2.17}$$

the total energy of the system is obtained from the variational principle as:

$$E = \min_{\vec{t}} \langle \mathrm{HF}|\, e^{-(T-T^\dagger)} H e^{T-T^\dagger}\, |\mathrm{HF}\rangle \tag{2.18}$$

while this ansatz is variational and spans the same Hilbert space as the original coupled cluster ansatz, Eq. (2.17) does not lead to equations which can be tractably solved on a classical computer[190,330]. To see this we can examine the BCH expansion of the similarity transform hamiltonian for UCC:

$$\begin{aligned} e^{T^\dagger-T} H e^{T-T^\dagger} =& H + [H,T] + \left[T^\dagger,H\right] + \frac{1}{2}([[H,T],T] \\ &+ \left[T^\dagger,\left[T^\dagger,H\right]\right] + \left[H,\left[T,T^\dagger\right]\right]) + \cdots \end{aligned} \tag{2.19}$$

In contrast with the expansion for CC (Eq. (2.16)), Eq. (2.19) involves terms that depend on the commutators between T and $T^\dagger$ operators, for which there is no natural termination point[190,330]. Therefore, the BCH series for UCC is infinite and thus there is currently no known method for efficiently evaluating the energy and amplitude equations on a classical computer without further approximation.

Nonetheless, the minimization of the UCC ansatz is of great interest to the quantum chemistry community that has been trying to develop tractable approximations to this theory for many

years[25,72,103,150,189]. Fortunately, the operator $e^{T-T^\dagger}$ can be readily applied on a quantum computer, which makes it possible to prepare UCC wavefunctions with truncated cluster expansions, as shown in[226,255,369].

2.3 VARIATIONAL QUANTUM EIGENSOLVER FOR UCC

The VQE algorithm comprises three iterative steps: 1) preparation of the wavefunction by application of parameterized state preparation unitaries; 2) determination of the expectation value of every term in the Hamiltonian via an efficient partial tomography[222] and 3) calculation of the total energy and determination of a new set of state preparation parameters in a classical computer. This scheme avoids the substantial overhead of quantum phase estimation that causes other quantum algorithms for chemistry to require very long coherent evolution. It also offers flexibility in the length of the circuit for state preparation, that depends on the choice of ansatz for the state preparation.

In the specific case of UCC, the preparation of the wavefunction encompasses two steps: preparation of the reference state, $|\Phi_0\rangle$, and application of the UCC unitary, $U(\vec{t})$, that prepares the UCC wavefunction. The algorithm starts with a guess of the UCC amplitudes, $\vec{t}^{\,(0)}$, and iteratively converges to a final set of parameter by variationally minimizing the energy. At the n-th iteration, the UCC wavefunction is prepared using $\vec{t}^{\,(n)}$ and the expectation value of the Hamiltonian, H, is obtained as the sum of the expectation values of all the terms, $\langle H\rangle = \sum_i \langle H_i\rangle$. The classical optimization routine produces a new estimate of the UCC amplitudes, $\vec{t}^{\,(n+1)}$. The algorithm convergences when the changes in both, total energy and $\vec{t}$, become smaller than suitable thresholds. In the following sections, we describe in detail the steps involved in the VQE implementation of the UCC ansatz. A graphical summary of the procedure is shown in Figure 2.1.

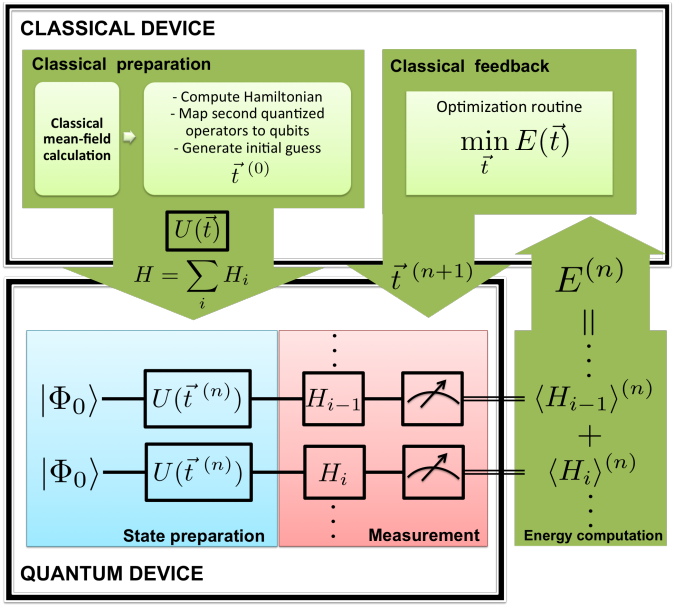


Figure 2.1: Schematic representation of the Variational Quantum Eigensolver algorithm applied to the UCC ansatz. The classical optimization routine adds the expectation values of the Hamiltonian terms to calculate the energy and estimates a new value for the coupled cluster amplitudes, $\vec{t}$. The process is repeated until achieving convergence on the total energy and $\vec{t}$.

2.3.1 Implementation of UCC on a quantum computer

To prepare the UCC ansatz on a quantum computer we need to map the UCC operator (Eq. (2.17)) onto operations that can be performed on the quantum computer. We start by rewriting the cluster operator as

$$U(\vec{t}) = e^{\Sigma_j t_j(\tau_j - \tau_j^\dagger)} \tag{2.20}$$

where τ_j represent an excitation operator and t_j the corresponding CC amplitude. Since excitation operators do not necessarily commute, the UCC unitary can be approximated using trotterization:

$$U\left(\vec{t}\right) \approx U_{Trot}\left(\vec{t}\right) = \left(\prod_j e^{\frac{t_j}{\rho}(\tau_j - \tau_j^\dagger)}\right)^\rho \tag{2.21}$$

where ρ is the trotter number. We emphasize that the goal of VQE is the approximation of the ground state of the problem Hamiltonian, and in that sense the purpose of Trotterizing UCC is to obtain a physically motivated ansatz that can be implemented on a quantum computer and not to approximate the exact UCC unitary. Correspondingly, the error of our ansatz is determined by the difference in energy with the ground state and not by the difference with the solution of the exact (non-Trotterized) UCC ansatz. In addition, these two quantities are expected to be different because the optimal amplitudes for both ansatzes do not necessarily coincide as a result of the variational optimization. This contrasts with algorithms for simulating Hamiltonian dynamics, where the goal is to approximate as closely as possible unitary time evolution and therefore the quality of the approximation is directly related to the Trotter error.

In this work we will employ the approximations with $\rho = 1$ and $\rho = 2$ as our state preparation unitaries. For $\rho = 1$:

$$U_1\left(\vec{t}\right)|\Phi_0\rangle = \prod_j e^{t_j(\tau_j - \tau_j^\dagger)}|\Phi_0\rangle \tag{2.22}$$

In the following section we will present numerical evidence that shows that these types of ansatzes are as effective as the one in Eq. 2.20. To implement Eq. 2.22 on a quantum computer, we need to map every unitary in the previous product to operations in the quantum computer. For this purpose we can use either the Jordan-Wigner (JW) or the Bravyi-Kitaev (BK) mappings [165,300,336] (See Appendix 2.6.2), obtaining:

$$(\tau_j - \tau_j^\dagger) = i\sum_k^{2^{2l_j-1}} P_k^j \tag{2.23}$$

where P_j^i represents a product of Pauli matrices with real coefficients and i is the imaginary unit. The index k runs over 2^{2l_j-1} products, where l_j is the excitation rank of the j-th excitation operator

τ_i (See Appendix 2.6.1). We will refer to each P_k^j in Eq. 2.23 as a *subterm*. For instance, a double excitation operator minus its complex conjugate will comprise eight subterms. Using the previous notation we can write:

$$U_1\left(\vec{t}\right) = \prod_j \exp\left(it_j \sum_k^{2^{2l_k-1}} P_j^k\right) \tag{2.24}$$

Furthermore, we can show that the subterms derived from the same $(\tau_j - \tau_j^\dagger)$ operator commute (See Appendix 2.6.1), which allow us to simplify the expression of the complex cluster unitary as follows:

$$U_1\left(\vec{t}\right) = \prod_j \prod_k^{2^{2l_k-1}} \exp(it_j P_k^j) \tag{2.25}$$

The terms in Eq. 2.25 can be implemented in a quantum computer using the digital model of quantum computation. In this chapter we will focus on the universal sets of gates typically employed for superconducting circuit (SQC) and trapped ion (TI) quantum computers[23,34]: single qubit rotations and CNOT or Mølmer-Sørensen (MS) gates, respectively. Thanks to their capabilities in number of qubits and coherent control, the SQC and TI architectures have allowed the first scalable demonstrations of digital quantum simulation[24,40,192].

Using the first set of gates, the exponentiation of a n-fold tensor product of Pauli-Z matrices can be done with $O(N)$ CNOT gates and a single single qubit (SQ) rotation. If there are Pauli-X or Y matrices in the tensor product we must apply the single-qubit Hadamard or $R_x(\frac{\pi}{2})$ gate to rotate to the X or Y basis, respectively, before we compute the parity of the set of qubits with CNOTs, and also apply the inverse gates as part of the uncomputing stage[300,336,359].

We point out that employing the BK transformation, the number of operations required for implementing a single $\tau_j - \tau_j^\dagger$ term scales as $O(\log(N))$[300], which represent a most advantageous

mapping when compared to the JW transformation that scales as $O(N)$. However, for architectures with limited connectivity (e.g. SQC), we will need extra SWAP operations to implement the exponentiation, which may eliminate the advantage of the BK transformation. In addition, there is recent evidence that the JW implementation is more robust to errors due to noise in the quantum computer, compared to BK[288], presumably because occupations are stored non-locally in the case of BK, which amplifies the impact of a local single qubit error on the representation of the fermionic operators. In contrast, each qubit stores a single occupation number in the JW representation.

The key for retaining a polynomial number of operations to perform VQE with a UCC ansatz is to truncate the CC expansion. A popular truncation in quantum chemistry is to consider only single and double excitations (UCCSD):

$$T \approx T_1 + T_2 \tag{2.26}$$

This approximation suffices to accurately describe many molecular systems and is exact for systems with two electrons. Employing UCCSD, the number of parameters grows as $\binom{N-\eta}{2}\binom{\eta}{2} + \binom{N-\eta}{1}\binom{\eta}{1} < O(N^2\eta^2)$ where N is the number of spin orbitals (mapped to qubits) and η the number of electrons in the system. Combining the scaling of the number of parameters with upper bounds for the number of gates required to implement a single parameter we can estimate upper bounds for the total number of operations involved in preparing the UCCSD ansatz for single iteration of the VQE algorithm. In the case of the BK transformation, the number of gates scales as $O(N^2\eta^2)$, up to logarithmic factors, compared to $O(N^3\eta^2)$ using the JW transformation. If non-local gates are available (e.g. in TI), the circuit depth for the JW implementation can be reduced by a factor of $O(N)$ using the ordering and parallelization techniques described in[137].

An alternative to CNOT gates, especially suited for ion trap architectures, is the

Mølmer-Sørensen (MS) gate[317,318], that has been previously employed in the context of quantum simulation of fermions[63,229]. Its unitary evolution can be represented by the sum over all joint rotations on qubit j and k of the register for an angle θ around an axis ϕ, which can be freely chosen:

$$U_{\mathrm{MS}}(\theta,\phi) = \exp\left(-i\frac{\theta}{2}\sum_{j<k}\sigma_j^{\phi}\sigma_k^{\phi}\right), \tag{2.27}$$

where $\sigma_j^{\phi} = \cos(\phi)\sigma_j^x + \sin(\phi)\sigma_j^y$. For $\theta = \pi/2$ and $\phi = 0$ the action of U_{MS} creates a fully entangled state under $\sigma^x\sigma^x$ operation. This non-local gate can be made to act on arbitrary subsets of qubits in various ways: (a) by spectroscopic decoupling of unwanted qubits from the interaction[289], (b) by selectively focussing laser beams on the desired qubits[86] or (c) the use of refocussing techniques[238].

Depending on the way in which the entangling operations on subregisters are implemented, this leads to a scaling of two entangling operations per parameter, largely reducing their number with respect to the implementation using CNOTs. This is a significant advantage as they remain the limiting factor in the current-day leading architectures, while single qubit operations can already be achieved with very high fidelities far beyond fault-tolerance thresholds. In addition, MS gates are particularly attractive when used with the Bravyi-Kitaev transformation, because the gate only needs to act on $O(\log N)$ qubits rather than $O(N)$ for the Jordan-Wigner transformation.

2.3.2 Choice and preparation of the reference state

In the limit of the complete cluster expansion, the UCC ansatz provides the exact solution for the many body problem. In practice, having a reference state with a high overlap with the exact wavefunction facilitates convergence[214]. Generally, the Hartree-Fock solution of the many-body

problem provides such reference. The Hartree-Fock state can be written as:

$$|\Phi_0\rangle = a_\eta^\dagger a_{\eta-1}^\dagger \dots a_1^\dagger |\rangle \tag{2.28}$$

where $|\rangle$ is the fermionic vacuum state. Using the molecular orbital basis, the Hartree-Fock state corresponds to a single product state in the computational basis after the BK or JW mappings are applied. For instance, in the JW mapping the HF state corresponds to the state $|0\rangle^{\otimes N-\eta} \otimes |1\rangle^{\otimes \eta}$, where the the single-particle basis is organized according to the one-particle energy from lowest to highest, the so-called canonical order. In this case the Hartree-Fock state can be constructed by initializing the qubit register with the first η qubits in $|1\rangle$ and $N-\eta$ in $|0\rangle$, which can be achieved with a single layer of single qubit rotations. In general, preparing a Hartree-Fock state in an arbitrary basis can be achieved using circuits of depth $O(N)$, even in quantum processors with limited connectivity[179].

In cases where the molecular wavefunction exhibits strong correlations, the Hartree-Fock state provides a poor starting guess. This problem arises in the description of bond breaking/formation, excited states, ground states of transition metal complexes, among other chemical phenomena[214], and can be helped by using a multireference approach. One possibility is to employ an entangled reference states obtained from a classical Multiconfigurational Self-Consistent Field (MCSCF) calculation[328] or a DMRG calculation with a small active space. As long as this state comprises of only a polynomial number of computational states, it can be prepared efficiently on a quantum computer combining the techniques proposed in[249,316,351] for preparing superpositions of Slater determinants with the most recent techniques for preparing individual Slater determinants[179]. Using these reference states, Eq. (2.17) can be applied without modification after redefining the space of virtual orbitals according to the occupation of each orbital, which can be determined by measuring the corresponding occupation-number operator. The UCC approach can be also

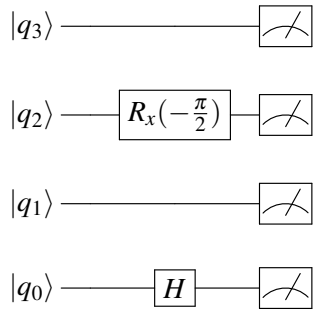


Figure 2.2: Circuit illustrating the measurement of the term $\sigma_3^z\sigma_2^y\sigma_1^z\sigma_0^x$ in the Z basis. We must apply H or $R_x(-\frac{\pi}{2})$ gates (or equivalent) to change basis when measuring Pauli-Y and Pauli-X operations.

extended to multireference cases by adopting an agnostic unitary coupled cluster ansatz, where the definition of the excitation operators is not linked to a specific reference state, as described in[226]. In this case, the indexes of the excitation operators in Eq. (2.7) run over the set of all possible spin-orbitals.

2.3.3 Energy measurement

Once the state preparation has been performed, the next step in the VQE algorithm is the calculation of the objective function that corresponds to the energy measurement $E = \langle\Phi_0|e^{-(T-T^\dagger)}He^{T-T^\dagger}|\Phi_0\rangle$. To avoid performing phase estimation, which has a prohibitively large circuit depth for current and near-future quantum devices, we employ the Hamiltonian averaging procedure, introduced in[222,226]. In this case the energy is calculated by measuring the expectation value of every term in the Hamiltonian and adding them to obtain the total energy:

$$E = \sum_i^M h_i\langle O_i\rangle \tag{2.29}$$

where every Hamiltonian term, O_i, comprises of a tensor product of Pauli matrices obtained from the JW or the BK transformations, multiplied by the corresponding Hamiltonian coefficient, h_i. The expectation value of a string of Pauli matrices, can be measured as illustrated in Figure 2.2

using projective measurements.

We can estimate the number of measurements required to converge the total energy to a precision ε following a frequentist approach, as shown in[353]. Assuming each term in the Hamiltonian is measured m_i times, the precision achieved in each term, ε_i, is given by:

$$\varepsilon_i^2 = \frac{|h_i|^2 \mathrm{Var}[\langle O_i \rangle]}{m_i} \tag{2.30}$$

where $\mathrm{Var}[\langle O_i \rangle]$ represents the variance of the expectation value of the operator O_i, which is upper-bounded by 1 in the case of Pauli terms. To achieve precision ε in the total energy we can choose $\varepsilon_i^2 = \frac{|h_i|}{\sum_j^M |h_j|}\varepsilon^2$. Taking into account the bound in the variances, we can estimate the total number of measurements, m, as:

$$m = \frac{\sum_j^M |h_j| \sum_i^M |h_i| \mathrm{Var}[\langle O_i \rangle]}{\varepsilon^2} \leq \frac{(\sum_j^M |h_j|)^2}{\varepsilon^2} \tag{2.31}$$

2.3.4 Parameter optimization

The final step of the VQE algorithm involves the minimization of the total energy with respect to the wavefunction parameters, that in the case of UCC correspond to the cluster amplitudes, $\vec{t}$. This is a non-linear optimization problem for which a variety of optimization algorithms has been proposed[367]. However, we note that in early demonstration of the VQE algorithm the objective function might exhibit a highly non-smooth character due to experimental noisy conditions. In this scenario, we might expect that direct search algorithms, which are more robust to noise, have an advantage over optimization methods that rely on gradients[183].

The optimization performance will also depend on the quality of the starting parameters. Fortunately, it is possible to generate starting guesses for the cluster amplitudes based on classical quantum chemistry approaches. For instance, classical CCSD employ the CC amplitudes obtained

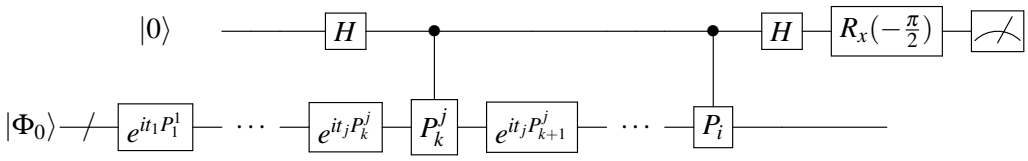


Figure 2.3: Circuit for measuring the imaginary part of $\langle\Phi_0|V_k^{j\dagger}(\vec{t})O_iU(\vec{t})|\Phi_0\rangle$ required in the calculation of the partial derivative $\frac{\partial E(\vec{t})}{\partial t_j}$. The $R_x(\frac{\pi}{2})$ gate rotates to the Y-basis.

from second order Møller-Plesset perturbation theory (MP2) as starting guesses to solve for the CC equations. The MP2 guess amplitudes are given by the equations:

$$t_i^a = 0; \quad t_{ij}^{ab} = \frac{h_{ijba} - h_{ijab}}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b} \tag{2.32}$$

where ε_p stands for the Hartree-Fock energy of the orbital p and h_{pqrs} represent the two electron integrals (Eq. (2.4)). This information is obtained directly from the Hartree-Fock calculation. As the solutions of truncated CC or truncated CI are also efficient, it is possible to use cluster amplitudes obtained from methods such as CCSD. One can easily compute both cluster amplitudes and MP2 amplitudes using modules provided in OpenFermion[225].

Classical approximations to the cluster amplitudes also serve as a criteria to reduce the number of parameters in the optimization. Before starting the VQE optimization, we can remove from the UCC unitary those excitation operators that have a small amplitude according to the classical estimate, as they are likely to also have a small contribution to the final wavefunction. Once the first optimization has been completed, we might include more excitation operators and repeat the optimization until a desired convergence threshold is achieved. The same strategy could be employed during the optimization process, discarding those operators for which the cluster amplitudes remain small after certain number of VQE iterations.

2.3.5 GRADIENT EVALUATION FOR UCC

Direct search algorithms can be more robust to noise than gradient-based approaches, but this generally comes at the cost of demanding a larger number of function evaluations to achieve convergence [183]. As the accuracy of quantum computers increases, the possibility of computing energy gradients in the quantum computer becomes more feasible. One possibility is to compute the gradient numerically, using for instance a finite difference formula. In this case, the accuracy of the gradient depends on the step size chosen, which would be limited by the precision of the experimental control over the parameters and by shot-noise limited measurements.

Alternatively, one might evaluate the gradient directly on the quantum computer given that an analytical implementation is available. Here we propose a method to compute the analytical gradient of the energy when a product of parametrized unitaries is employed in the state preparation.

Consider a unitary ansatz analogous to the one defined in Eq. (2.25):

$$U\left(\vec{t}\right) = \prod_{j}^{N_P} \prod_{k}^{N_S^j} \exp(ic_k^j t_j P_k^j) \tag{2.33}$$

where N_P stands for the number of parameters and N_S^j stands for the number of subterms that depend on the j-th parameter. P_k^j is a string of Pauli matrices. c_k^j is a constant, that in the case of the UCC ansatz corresponds to the constant factors obtained arising from the mapping of fermionic operators to qubit operators. Consider the state $\Psi\left(\vec{t}\right)$, prepared as $\Psi\left(\vec{t}\right) = U\left(\vec{t}\right)|\Phi_0\rangle$, where $|\Phi_0\rangle$ is a reference wavefunction that do not depend on $\vec{t}$. Also consider a Hamiltonian, H, which is independent of the parameters $\vec{t}$. In this case, the derivative of the expectation value of the energy,

$E(\vec{t}) = \langle \Psi(\vec{t}) | H | \Psi(\vec{t}) \rangle$, with respect to the parameter t_j will be given by

$$\frac{\partial E(\vec{t})}{\partial t_j} = \langle \Phi_0 | U^\dagger(\vec{t}) H \frac{\partial U(\vec{t})}{\partial t_j} | \Phi_0 \rangle + \langle \Phi_0 | \frac{\partial U(\vec{t})^\dagger}{\partial t_j} H U(\vec{t}) | \Phi_0 \rangle \tag{2.34}$$

$$= i \sum_k^{N_S^j} \langle \Phi_0 | U^\dagger(\vec{t}) H V_k^j(\vec{t}) | \Phi_0 \rangle - \langle \Phi_0 | V_k^{j\dagger}(\vec{t}) H U(\vec{t}) | \Phi_0 \rangle \tag{2.35}$$

$$= 2 \sum_k^{N_S^j} c_k^j \, \mathrm{Im}(\langle \Phi_0 | V_k^{j\dagger}(\vec{t}) H U(\vec{t}) | \Phi_0 \rangle) \tag{2.36}$$

Where the operator $V_k^j(\vec{t})$ is defined as the unitary of Eq. (2.33) but with the operator P_k^j interleaved between the unitaries $\exp(it_j P_{k-1}^j)$ and $\exp(it_j P_k^j)$. Explicitly:

$$\begin{aligned} V_k^j(\vec{t}) = & \exp(it_j P_1^1) \cdots \exp(it_j P_{k-1}^j) P_k^j \exp(it_j P_k^j) \\ & \exp(it_j P_{k+1}^j) \cdots \exp(it_{N_P} P_{N_S^{N_P}}^{N_P}) \end{aligned} \tag{2.37}$$

Combining Eq. (2.36) with the decomposition of the Hamiltonian in Eq. (2.29), we obtain a working expression for computing $\frac{\partial E(\vec{t})}{\partial t_j}$:

$$\frac{\partial E(\vec{t})}{\partial t_j} = 2 \sum_i^M h_i \left(\sum_k^{N_S^j} c_k^j \, \mathrm{Im}(\langle \Phi_0 | V_k^{j\dagger}(\vec{t}) O_i U(\vec{t}) | \Phi_0 \rangle) \right) \tag{2.38}$$

We can evaluate the imaginary part of $\langle \Phi_0 | V_k^{j\dagger} H_i U(\vec{t}) | \Phi_0 \rangle$ with the circuit of Figure 2.3, which follows the approach introduced in Ref.[249] to measure expectation values of the form $\langle V^\dagger U \rangle$. We use a state register initialized with the reference state tensor an ancilla qubit initialized in $|+\rangle$. First, we apply the unitaries of Eq. (2.33) to the state register up to $\exp(it_j P_k^j)$, after which we apply the operator P_k^j controlled by the ancilla qubit. Subsequently, we apply the remaining unitaries to the state register, followed by the local operator H_i controlled by the ancilla qubit.

Finally, we apply a Hadamard gate in the ancilla qubit to obtain the state

$$\frac{|0\rangle \otimes (U|\Phi_0\rangle + O_i V_k^j(\vec{t})|\Phi_0\rangle) + |1\rangle \otimes (U|\Phi_0\rangle - O_i V_k^j(\vec{t})|\Phi_0\rangle)}{2} \tag{2.39}$$

The imaginary part of $\langle\Phi_0|V_k^{j\dagger}(\vec{t})O_iU(\vec{t})|\Phi_0\rangle$ can be recovered by measuring the ancilla qubit in the Y-basis. The variance of the j-th component of the gradient as computed with the circuit of Figure 2.3 will be given by:

$$\mathrm{Var}\left[\frac{\partial E(\vec{t})}{\partial t_j}\right] = 4\sum_i^M |h_i|^2 \sum_k^{N_S^j} |c_k^j|^2 \mathrm{Var}\left[\langle\sigma^y\rangle_{O_i,P_k^j}\right] \tag{2.40}$$

where

$$\langle\sigma^y\rangle_{O_i,P_k^j} = \left\langle 0\otimes\Phi_0 \left| C^\dagger_{O_i,P_k^j}(\sigma^y\otimes I)C_{O_i,P_k^j} \right| 0\otimes\Phi_0 \right\rangle \tag{2.41}$$

and C_{O_i,P_k^j} represents the circuit for gradient estimation for the subterm P_k^j and the observable O_i. To estimate the number of measurements required to achieve precision $\tilde{\varepsilon}_j$ in the j-th component of the gradient, we will first consider the number of measurements required to estimate the contribution of the circuit C_{O_i,P_k^j} to precision $\tilde{\varepsilon}_{j,k}^i$:

$$\tilde{m}_{j,k}^i = \frac{|c_k^j|^2 \mathrm{Var}\left[\langle\sigma^y\rangle_{O_i,P_k^j}\right]}{(\tilde{\varepsilon}_{j,k}^i)^2} \tag{2.42}$$

For the UCC ansatz, the constants $c_{j,k}^i$ have the same norm, $|c_{j,k}^i| = |c_j|$ and fulfill $\sum_k^{N_S^j} |c_{j,k}^i| = 1$. Therefore we can choose $(\tilde{\varepsilon}_{j,k}^i)^2 = |c_j|(\tilde{\varepsilon}_j^i)^2$, where $\tilde{\varepsilon}_j^i$ is the precision for estimating the contribution of the operator O_i to the gradient variance. In addition, we can apply the same sampling strategy chosen for estimating the energy (Eq. (2.31)), and choose $(\tilde{\varepsilon}_j^i)^2 = \frac{|h_i|\tilde{\varepsilon}_j^2}{\sum_l^M |h_l|}$.

Introducing these considerations into Eq. (2.42), we obtain:

$$\tilde{m}_j = \left(4\sum_l^M |h_l|\right) \frac{\sum_i^M \sum_k^{N_S^j} |h_i||c_k^j| \mathrm{Var}\left[\langle \sigma^y \rangle_{O_i,P_k^j}\right]}{\tilde{\varepsilon}_j^2} \tag{2.43}$$

We can get an upper bound to Eq. (2.43) by considering the upper bound of the variance and including the properties of the coefficients $c_{j,k}^i$:

$$\tilde{m}_j \leq 4\frac{\left(\sum_i^M |h_i|\right)^2}{\tilde{\varepsilon}_j^2} \tag{2.44}$$

For comparison, consider the simplest central finite difference formula that requires two energy evaluations to compute each gradient component:

$$\frac{\partial E(\vec{t})}{\partial t_j} \approx \frac{E(t_1,..,t_j+\delta,..,t_{N_P}) - E(t_1,..,t_j-\delta,..,t_{N_P})}{2\delta} \tag{2.45}$$

where δ is the step size. As in the case of the analytical gradient, we choose to estimate the j-th gradient component to precision $\tilde{\varepsilon}_j$. The precision in the numerical gradient will depend on the precision of the numerator and denominator of Eq. (2.45). Assuming no error in the denominator and a non-zero numerator, the precision for estimating the energies in the numerator, ε_j, can be chosen to guarantee that the relative precisions of the gradient component and the numerator are the same. This condition requires $\varepsilon_j = \frac{2\delta\tilde{\varepsilon}_j}{\sqrt{2}}$. Combining this requirement with Eq. (2.31), we can bound the number of measurements for estimating the j-th component of the gradient as:

$$\tilde{m}_j \leq 4\left(\frac{(\sum_i^M |h_i|)^2}{(2\delta)^2\tilde{\varepsilon}_j^2}\right), \tag{2.46}$$

where the estimate considers two energy evaluations per gradient component. To achieve precision $\tilde{\varepsilon}$ in the norm of the gradient, we could choose $\tilde{\varepsilon}_j^2 = \frac{\tilde{\varepsilon}^2}{N_P}$, allowing us to bound the sampling cost of

gradient approximations as:

$$\tilde{m} \leq CN_P \left(\frac{(\sum_i^M |h_i|)^2}{\tilde{\varepsilon}^2} \right), \tag{2.47}$$

where $C = \frac{4}{(2\delta)^2}$ for the simplest central difference formula and $C = 4$ for the analytical gradient estimated using Figure 2.3. The same bounds can be derived for the UCC approximations with more than one Trotter step, $\rho > 1$. In this case, the factor $\frac{1}{\rho}$ appears multiplying the constants $c^i_{j,k}$, but the number of circuits contributing to N_S^j also increases by factor of ρ, canceling out the $\frac{1}{\rho}$ factor in the estimation of the bound.

The previous analysis indicates that the sampling cost of the numerical gradient increases quadratically with decreasing the step size. From the analysis of Eq. (2.47), we expect that for $\delta < 0.5$ the numerical gradient will have a larger sampling cost than the analytical gradient approach. In addition, the accuracy of the numerical gradient depends on the step size used in the central difference formula and sets a lower bound to the precision that can be obtained from the numerical gradient.

From Eq. (2.47), we also conclude that the gradient estimation is more expensive than estimating the energy by a factor proportional to the number of parameters. However, the relative cost of gradient-based and gradient-free optimization is ultimately determined by the number of iterations required for convergence. Usually, gradient based methods employ a number of gradient evaluations much smaller than the number of energy evaluations employed by derivative-free methods.

Finally, we point out that the sampling cost can be reduced by adapting the precision required in each optimization step according to the norm of the gradient, instead of employing a fixed gradient precision throughout the optimization. With this strategy, the first steps would require less measurements compared to the final steps, where the gradient norm is smaller.

2.3.6 VQE-UCC WITH AN ACTIVE SPACE APPROXIMATION

Several approximations that have been designed to reduce the computational cost of classical quantum chemistry algorithms can be extrapolated to the quantum implementation. A particular strategy that could be exploited to reduce the number of quantum resources for a VQE-UCC calculation is the *complete active space* (CAS) approach[281]. The CAS approximation consists in dividing the orbital space into a set of *inactive* (I) and *active* (A) orbitals such as the occupation of the orbitals in the inactive space remains unchanged. This idea exploits the fact that for most of the quantum chemistry Hamiltonians, including those cases with a strong multireference character, the wavefunction is qualitatively dominated by a relatively small number of Slater determinants that can be effectively captured by expanding the wavefunction in a subspace defined by the active orbitals.

In most quantum chemistry applications, the CAS approximation is employed to treat static correlation effects, meaning that a relatively small active space is selected to obtain a qualitatively correct wavefunction that serves as reference state for further perturbation theory or Coupled Cluster refinements[214,328]. Nonetheless, one might also consider the choice of an active space that is sufficiently large such as both static and dynamical correlation effects can be described up to certain accuracy.

In the case of the CAS approximation applied to single reference UCC, one selects an active space comprised of η_A electrons distributed among N_A spatial orbitals. This choice of active space is denoted as $CAS(\eta_A, N_A)$. The active orbitals usually correspond to a selection of the highest occupied orbitals and the lowest virtual orbitals. The cluster operators are then redefined such as excitations are only allowed among active orbitals,

$$T^A = \sum_i^{\eta_A} T_i \tag{2.48}$$

Considering the separation between active and space orbitals, we can rewrite the reference state as $|\Phi_0\rangle = |\Phi_0^A\rangle \otimes |\Phi_0^I\rangle$, where $|\Phi_0^I\rangle$ and $|\Phi_0^A\rangle$ are defined over the inactive space and active space, respectively. Consequently we can write the total energy as:

$$E = \langle \Phi_0^A | e^{-(T^A - T^{A\dagger})} \tilde{H}^A e^{T^A - T^{A\dagger}} | \Phi_0^A \rangle \tag{2.49}$$

where the effective Hamiltonian $\tilde{H}^A$ is obtained by evaluating the following expression:

$$\tilde{H}^A = \langle \Phi_0^I | H | \Phi_0^I \rangle \tag{2.50}$$

Since the reference state corresponds to a product state, the calculation of the effective Hamiltonian can be performed efficiently on a classical computer. In this case, we can obtain an approximate solution to the VQE problem by performing a VQE-UCC calculation for the effective active space Hamiltonians, $H^A_{jj'}$. The CAS-UCC approach reduces the number of qubits required for a calculation by a factor of N_A/N. Similarly, the number of parameters for the preparation of the UCCSD wavefunction is reduced by a factor of $(\eta_A N_A)^2/(N\eta)^2$ with respect to full-UCCSD, as the scaling becomes $O({\eta_A}^2 N_A^2)$.

A number of strategies for selecting active spaces to describe static correlation have been proposed in the context of quantum chemistry. Generally, these strategies employ the occupation of approximate natural orbitals, which are the orbitals that diagonalize the one particle density matrix, as a criteria to choose the active space. Orbitals with integer occupation are generally discarded, and only those with fractional occupation within certain thresholds are considered as part of the active space. The approximate one particle density matrix is obtained from methods that include some amount of correlation and that are relatively inexpensive, such as MP2[159]. Another commonly used approach employs the unrestricted natural orbitals (UNO) obtained from

unrestricted Hartree-Fock calculations[5,174]. More recently, a scheme based on entanglement measurement among orbitals has been also proposed[323].

We can take advantage of one of these approaches to define an initial active space in a suitable basis for the UCC calculation. The solution obtained with the initial active space can be employed as an initial guess for another CAS-UCC calculation with a larger active space. This process can be repeated until the simulation is performed on the entire basis, in which case we expect the algorithm to converge faster as in each iteration a better approximation to the exact UCC wavefunction is obtained. One can also stop the optimization after the energy does not improve beyond a pre-defined threshold. In the later case, we also achieve a reduction in the number of qubits required for the calculation.

2.4 Numerical assessment

2.4.1 Classical simulation of VQE-UCC

To illustrate the algorithmic details of the the scalable VQE-UCC algorithm, we simulated the VQE-UCC calculation of small molecules. The molecular integrals were obtained using the PSI4 package[292] and the molecular Hamiltonian was mapped using the Jordan-Wigner transformation. The UCC unitary was constructed with a truncated cluster operator and the symbolic representation was transformed into unitaries comprising strings of Pauli matrices, following the same procedure employed for the Hamiltonian. To assist these transformations, we employed the OpenFermion (`www.mcclean2017openfermion.org`) library[225].

The simulation of the circuit proceeds by calculation of the UCC wavefunction from the the matrix representation of the UCC unitary and the reference state. The optimization was performed using three direct search algorithms available in the *scipy.optimize* library, namely the Nelder-Mead[242], Powell[266] and COBYLA[267] algorithms. We also employed the L-BFGS-B

method[55] with numerical gradients for comparison, using the central finite difference formula (Eq. (2.45)). The energy and parameter thresholds for convergence were fixed at 10^{-5} a.u 10^{-4} a.u respectively. For the L-BFGS-B algorithm, the convergence threshold for the projected gradient was fixed at 10^{-4}. In all cases the maximum number of function evaluations was fixed to 20,000. Finally, we point out that all our numerical experiments assume that function evaluations are performed in double precision arithmetic, unless indicated otherwise.

2.4.2 VQE-UCC RESULTS FOR H_4 MOLECULAR SYSTEMS

A practical and informative assessment of quantum chemistry simulation involves the study of chemical transformations, such as bond-breaking, isomerization or configurational changes. These processes are generally described through scanning geometries along certain directions of a PES. Along the PES, the amount of entanglement of the wavefunction varies greatly and this impacts the performance of the ansatz employed to approximate the wave function.

In order to illustrate these aspects, we selected a model in which the amount of entanglement in the wavefunction can be continuously varied and which is simple enough to enable simulations. We have considered the PES of a system comprising four hydrogen atoms investigated along three different paths: rectangular (R), trapezoidal (T), and linear (L), as described in Figure 2.4. These systems have been widely employed by the quantum chemistry community as a benchmark for multireference methods[158,214]. We studied 19 different geometries for the trapezoidal path generated by varying the parameter θ between $90 and 180$. For the linear and the parallel paths, we studied 24 different geometries generated by varying the parameter r between 0.6Å and 5.0Å.

2.4.2.1 INFLUENCE OF THE OPTIMIZATION METHOD IN THE VQE PERFORMANCE

We evaluated the effectiveness of the strategies proposed to generate the initial guess for the cluster amplitudes and optimization methods based on three criteria:

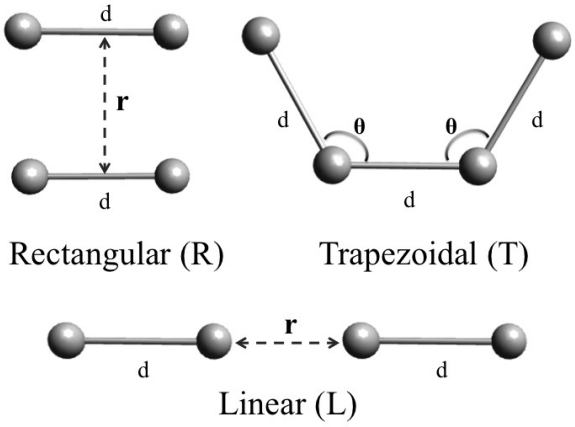


Figure 2.4: Description of geometries for the H_4 model systems studied in this work. The potential energy surfaces are defined as a function of the variable r for the rectangular (R) and linear (L) geometries and as a function of θ for the trapezoidal (T) geometry. The value of the parameter d is kept fixed at 2.0 Å.

1. the error in the calculated energy with respect to the FCI solution, $E_{\mathrm{FCI}} - E_{\mathrm{VQE}}$,

2. the accuracy of the wavefunction evaluated as the infidelity $(1 - |\langle \Psi_{\mathrm{VQE}} | \Psi_{\mathrm{FCI}} \rangle|^2)$ and

3. the number of function evaluations required for convergence.

We compared the four optimization methods described in the previous section using three different starting guesses:

1. random, in which random values are chosen uniformly in the interval -0.25 to 0.25,

2. starting with all the amplitudes set to zero, which corresponds to using the Hartree-Fock solution as initial guess and

3. the MP2 approximation to the cluster amplitudes.

The full optimization is comprised of a total of 52 parameters. To evaluate the performance of the random guess approach we ran the algorithm 10 times and averaged the results.

Figure 2.5 compares the average number of function evaluations and energy error along the rectangular, trapezoidal, and linear paths of the H_4 system. We observe that the Nelder-Mead and

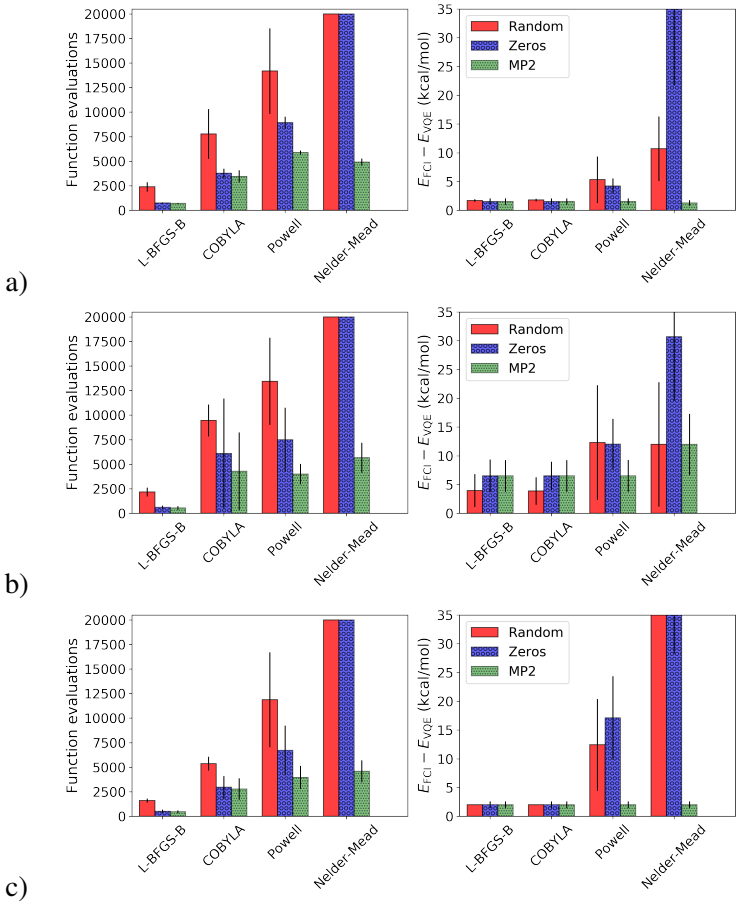


Figure 2.5: Average performance of the VQE algorithm applied to the H_4 system along the a) trapezoidal b) linear and c) parallel paths using four optimization methods (L-BFGS-B, COBYLA, Powell and Nelder-Mead) and three different methods to initialize the parameters: Randomly (Random), set to zero (Zeros) and set to the MP2 amplitudes (MP2). We compare the number of function evaluation required for convergence (left panel) and the error in the final energy with respect to the FCI solution (right panel). Error bars indicate one standard deviation. The range in the energy plots is truncated to 35 kcal/mol to facilitate comparison.

the Powell methods exhibit a high variability in their performances when the parameters are initialized at zero or randomly, as indicated by the large standard deviations in the wavefunction accuracy. In particular, Nelder-Mead fails to converge in less than 20000 function evaluations and performs poorly, with energy errors beyond 10 kcal/mol and overlaps with the exact wavefunction below 0.8. The Powell method has a better performance in the number of function evaluations but is still outperformed by L-BFGS and COBYLA. On the other hand, the COBYLA and the L-BFGS-B methods converge to almost the same minimum for most of the points of the PES,

independent of the method employed to generate the initial guess. This is indicated by the much larger energy accuracies compared to the results of Nelder-Mead and Powell.

The use of the MP2 guesses for the cluster amplitudes significantly reduces the number of function evaluations for all the optimization methods as observed in the left panel of Figure 2.5. MP2 guesses also improve the average accuracy of the energy obtained with the Nelder-Mead and Powell methods, as observed in the right panel of Figure 2.5. We point out that for systems that experience strong correlation the MP2 amplitudes might be a poor starting point, although still better than the random or zeros guesses. In those cases, more reliable methods such as Density Matrix Renormalization Group (DMRG) with a small active space and a small bond dimension could provide better initial guesses at the expense of classical computation time[340]. These results illustrate how the incorporation of classical approaches can improve the performance of quantum simulation by providing physically meaningful starting guesses and also highlight the importance of the choice of the optimization method.

2.4.2.2 Effect of trotterization in the optimization

Table 2.1 compares the performance of the trotterized UCC ansatz (Eq. (2.21)) using 1 and 2 trotter steps with the performance of the non-trotterized ansatz (Eq. (2.20)). For these calculations we employed the COBYLA and the L-BFGS-B optimization methods with the MP2 guess. To measure the quality of the results we use the average infidelity with respect to the FCI wavefunction as well the non-paralellism error (NPE). The NPE is calculated according to the formula:

$$\mathrm{NPE} = \max(E_{\mathrm{UCCSD}} - E_{\mathrm{FCI}}) - \min(E_{\mathrm{UCCSD}} - E_{\mathrm{FCI}}) \tag{2.51}$$

which quantifies the maximum error obtained when computing energy differences between points in the PES using the UCCSD approach. As observed in Table 2.1, the quality of the results obtained with the trotterized unitaries is almost identical to the that of the exact implementation of Eq. (2.20) when using the L-BFGS-B optimization method. We also notice that the approximation with 2 trotter steps converges faster in average than the unitary with only one trotter step.

Using COBYLA, the trotterized unitaries produce results similar to those obtained with Eq. (2.20) for the trapezoidal and the parallel paths. In contrast, COBYLA exhibits a lower average performance for the linear system, as shown in Table 2.1. A better insight into this result is offered by Figure 2.6, where we plot the error in the wavefunction along r for the linear path as computed with the COBYLA and the L-BFGS-B methods. The error in the wavefunction is quantified as the difference between 1.0 and the absolute value of the overlap of the UCCSD and the FCI wavefunctions. We observe that the COBYLA algorithm provides wavefunctions with overlaps below 0.95 and as low as 0.78 between $0.8 - 1.6$ Å, which corresponds to a section of the PES with strong multireference character. For these geometries, the COBYLA algorithm reaches the maximum number of functions evaluations when using Eq. (2.21) with 1 trotter step. Increasing the number of trotter steps seems to partially alleviate this problem. In contrast, a gradient based approach such as L-BFGS-B provides better results in the $0.8 - 1.6$ Å range. Interestingly, as the distance increases beyond 2.6 Å, the difference between the trotterized and the exact unitary becomes more prominent. We point out, however, that in all these cases the overlap is larger than 0.999 with a single trotter step.

Table 2.1: Comparison of the VQE results obtained with the ansatz from Eq. (2.20) and Eq. (2.25) with one and two trotter steps, for the calculation of the PES of the H_4 systems. We compared the average overlap with the FCI wavefunction, non-parallelism error and average number of function evaluations along the trapezoidal, parallel and linear paths of the H_4 system, obtained using the L-BFGS-B and COBYLA optimization methods. The molecular Hamiltonian was obtained with a STO-6G basis set.

Optimization method	System	Approximation	Average Overlap	NPE in PES* (kcal/mol)	Number of energy evaluations
L-BFGS-B	Trapezoidal	Eq. (2.25) $\rho = 1$	0.994 ± 0.006	1.4	861 ± 73
		Eq. (2.25) $\rho = 2$	0.995 ± 0.005	1.5	615 ± 32
		Eq. (2.20)	0.995 ± 0.005	1.5	703 ± 51
	Parallel	Eq. (2.25) $\rho = 1$	0.996 ± 0.008	2.0	590 ± 144
		Eq. (2.25) $\rho = 2$	0.997 ± 0.007	2.0	436 ± 149
		Eq. (2.20)	0.997 ± 0.006	2.0	467 ± 142
	Linear	Eq. (2.25) $\rho = 1$	0.998 ± 0.006	7.1	710 ± 99
		Eq. (2.25) $\rho = 2$	0.999 ± 0.005	6.9	487 ± 158
		Eq. (2.20)	0.999 ± 0.005	6.5	534 ± 182
COBYLA	Trapezoidal	Eq. (2.25) $\rho = 1$	0.994 ± 0.006	1.0	3703 ± 1023
		Eq. (2.25) $\rho = 2$	0.995 ± 0.005	1.5	2753 ± 340
		Eq. (2.20)	0.995 ± 0.005	1.5	3468 ± 622
	Parallel	Eq. (2.25) $\rho = 1$	0.998 ± 0.006	2.0	2431 ± 857
		Eq. (2.25) $\rho = 2$	0.999 ± 0.005	2.0	2047 ± 665
		Eq. (2.20)	0.999 ± 0.005	2.0	2820 ± 1086
	Linear	Eq. (2.25) $\rho = 1$	0.968 ± 0.068	5.1	5115 ± 5475
		Eq. (2.25) $\rho = 2$	0.990 ± 0.030	6.9	2882 ± 3620
		Eq. (2.20)	0.997 ± 0.006	6.5	4231 ± 3880

* Non-parallelism error.

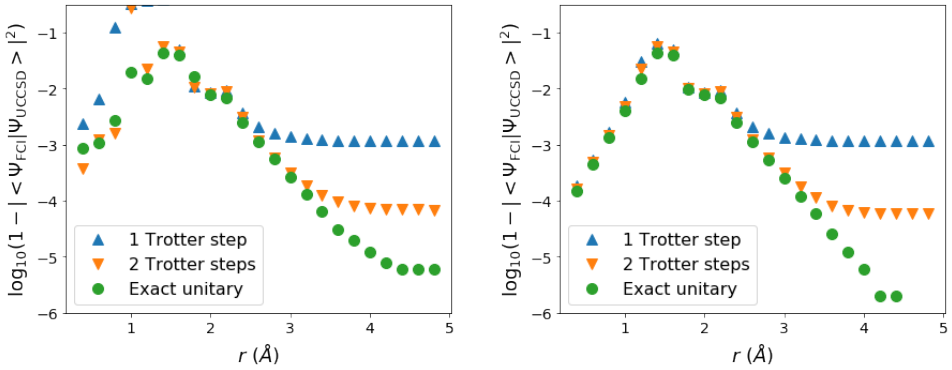


Figure 2.6: Comparison between the error in the wavefunctions obtained using a) COBYLA and b) L-BGFG-B optimizations along the linear path of the H_4 system. The UCCSD wavefunctions were obtained using the exact UCC unitary (Eq. (2.20), red dots) and the trotterized version (Eq. (2.25)) with one trotter step (blue triangles up) and two trotter steps (green triangles down). The error in the wavefunction is quantified as $1 - |\langle \Psi_{UCCSD}|\Psi_{FCI}\rangle|$.

2.4.2.3 Reduction in the number of parameters by pre-screening of cluster amplitudes

Classical approximations can provide a criterion to discard excitation operators with small amplitudes, which have a minor contribution to the wavefunction expansion. MP2 amplitudes, for instance, provide an approximation of the amplitudes of double excitation operators, which are responsible for the scaling of the number of parameters in the UCCSD method as a function of the system size. Given the set of MP2 amplitudes, $\{t_{ij}^{ab(\mathrm{MP2})}\}$, we can discard all the excitation operators such as $|t_{ij}^{ab(\mathrm{MP2})}| < d$, where d is a suitable threshold. Table 2.2 displays the average performance of UCCSD calculations in the H_4 systems using a reduced number of parameters for different values of d.

Table 2.2: Results of VQE calculations for the H_4 systems using prescreening with the MP2 amplitudes. We compared the results obtained for different values of the threshold (d) with the calculation including all the parameters. The L-BFGS-B optimization method was used. For $d < 10^{-3}$ the results are the same as for $d = 10^{-3}$.

System	Number of parameters			Max. difference in PES * (kcal/mol)		Energy evaluations x 10^3		
	$d = 10^{-2}$	$d = 10^{-3}$	All	$d = 10^{-2}$	$d = 10^{-3}$	$d = 10^{-2}$	$d = 10^{-3}$	All
Trapezoidal	24	26	52	<7x10-4	<7x10-4	1.17±0.11	1.20±0.13	3.5±0.6
Parallel	24	26	52	0.07	<7x10-4	1.12±0.44	1.17±0.43	2.8±1.0
Linear	24	26	52	0.76	0.20	1.26±0.43	1.37±0.37	4.2±3.8

* Maximum difference between the PES calculated with all the parameters and the PES obtained from the pre-screened calculation.

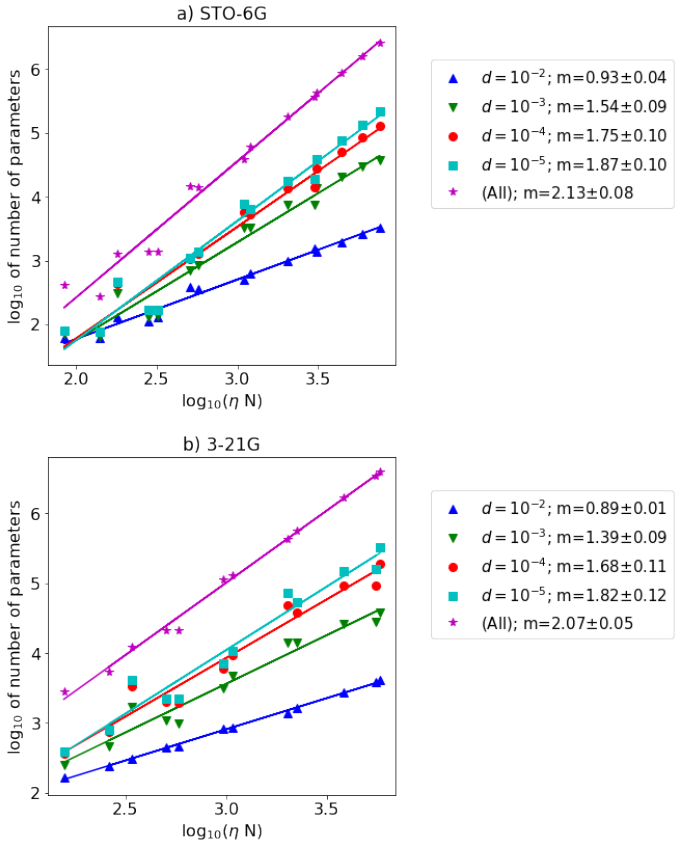


Figure 2.7: Number of parameters in the VQE unitary for different values of the threshold d for some molecules. We employed two different basis sets: a) STO-6G and b) 3-21G. The results are plotted against the product of the number of basis set functions and electrons, ηN. The correlation coefficient (R) and slope (m) of the linear regressions are shown in the legend. The list of molecules include: hydrocarbons (C1-C8), BeH_2, Benzene, N_2, O_2, B_2H_6, ethanol and water.

For the H_4 systems, only 10 out of the 34 double excitation operators have a significant effect on the total energy. The errors in the energy associated to the discarded parameters are in all cases smaller than chemical accuracy. Via this pre-screening process, we discard excitation operators that are expected to have small amplitudes and therefore small contributions to the accuracy of the wavefunction. Due to the elimination of non-critical excitation operators, the number of gates and thereby the circuit depth is reduced when compared to an implementation of the full, non-truncated UCC ansatz. The smaller number of parameters also facilitates the convergence of the optimization method. For the H_4 systems, the number of function evaluations decreases by a factor of 3.

To gain insights into the effect of the screening in the scaling of the number of parameters for UCC, we applied our reduction strategy to a series of small molecules for different values of the threshold d. The results are shown in Figure 2.7 as a function of the product of the number of electrons and the number of orbitals of the system, $N\eta$. We observe that the number of parameters, and consequently the depth of the VQE circuit, decreases by almost one order of magnitude using a threshold of 10^{-5}. In addition, for thresholds above 10^{-5}, it is also possible to achieve a subquadratic scaling in $N\eta$, compared to the formal quadratic scaling of the full UCCSD ansatz.

2.4.2.4 GRADIENT BASED OPTIMIZATION

Finally, we studied the performance of UCC optimizations with the analytical and numerical gradient approaches proposed in Section 2.3.5. First, we compared the sampling cost of the analytical and numerical gradient via numerical simulation. We calculated the error in the estimated gradient, Δg, as a function of the number of samples employed in the gradient estimation. The gradient error is quantified as the norm of the difference between the estimated gradient, $\tilde{g}$, and the exact gradient, g, $\Delta g = ||\tilde{g} - g||_2$. In our numerical simulations, the exact gradient corresponds to the analytical gradient computed to machine precision. To compute the number of measurements, we employed the equality of Eq. (2.31) for the numerical gradient and Eq. (2.43) for the analytical one.

Figure 2.8 illustrates the behavior of the error in the gradient as a function of the number of measurements for a single instance of the H_4 system in a linear configuration. Each point in the plot corresponds to an average over 100 gradient estimations for randomly sampled amplitudes. We compare the numerical gradient with different values of the step size, δ, and the analytical gradient. For $\delta = 0.5$ and high error rates, the numerical gradient has a sampling cost comparable to the analytical gradient. However, increasing the sampling cost beyond 10^8 does not further improve the quality of the numerical approach as the the method reaches its accuracy limit. A

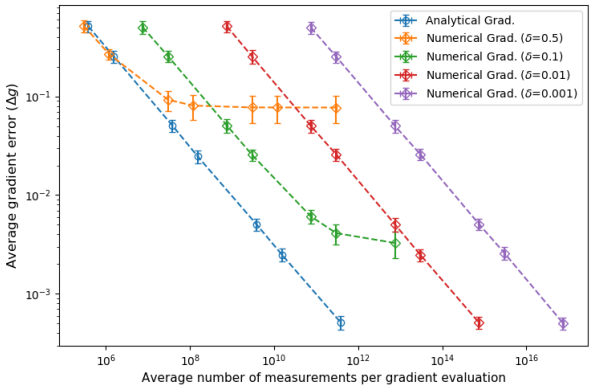


Figure 2.8: Average error of the numerical gradient as a function of the number of measurement employed for the gradient estimation. We compare the analytical gradient and the the numerical gradient for several step sizes. Averages were calculated over 100 random amplitudes drawn uniformly from the interval $[0, 2\pi]$ for the linear H_4 system with $r = 1.2$. Error bars correspond to one standard deviation.

similar behavior is observed for the numerical gradient with $\delta = 0.1$ for errors below 10^{-3}. Using smaller step sizes guarantees the same error rate achieved with analytical gradient but with an exceedingly larger sampling cost. Our results confirms the analysis presented in Section 2.3.5, indicating that the analytical gradient might be order of magnitudes cheaper than numerical gradients in experimental realizations of VQE.

To further understand the relative performance of the analytical and numerical gradients, we numerically simulated the impact of control errors on these methods. Control errors refer to the difference between the formal specification of a variational circuit $U(\vec{p})$, and the actual operation that this specification effects on the quantum computer, $\tilde{U}(\vec{p})$. For simplicity, we will model the control errors as $\tilde{U}(\vec{p}) = U(\vec{p} + \Delta\vec{p})$. In our numerical simulations, $\Delta\vec{p}$ is described as a normal random variable with standard deviation $\Delta\Theta$.

Figure 2.9 shows the magnitude of Δg for the analytical and the numerical gradients as a function of the parameter $\Delta\Theta$. We fixed the value of δ such as the contribution of the control errors is dominant in the numerical gradient for the ranges of $\Delta\Theta$ explored. Our results show that Δ_g scales linearly with $\Delta\Theta$, in contrast with the quadratic scaling in δ. In experimental

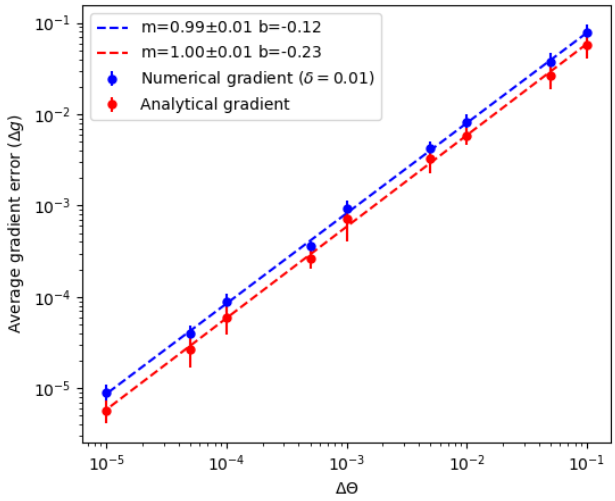


Figure 2.9: Average error of the numerical gradient as a function of the standard deviation of control errors in the quantum circuit ($\Delta\Theta$). m and b correspond to the slope and intercept of the linear regression. Averages were calculated over 100 random amplitudes drawn uniformly from the interval $[0, 2\pi]$ for the linear H_4 system with $r = 1.2$. The same scaling was observed for other instances of the H_4 system.

implementations of VQE, $\Delta\Theta$ imposes a practical lower bound to the value of δ employed in the estimation of the numerical gradient and consequently the contribution of control errors will dominate Δg.

Table 2.3: Average error in the UCC energy (hartrees) and average number of gradient calls employed for the optimization using analytical and numerical gradients (with $\delta = 0.05$ and $\delta = 0.10$) under the effect of control errors. The error in the energy corresponds to the difference between the optimal energy obtained for 150 VQE runs under control noise and the optimal value for the noiseless optimization with the analytical gradient. All the calculations employed a trotterized ansatz with one trotter step and the same stopping criteria for L-BFGS-B. The UCC amplitudes were initialized with the MP2 amplitudes. The parameter $\Delta\Theta$ was fixed to 0.01. The calculations were performed for instances of the trapezoidal, linear and parallel H4 system with the UCC ansatz ($r = 1.2$ and $\theta = 135.0^o$).

	Trapezoidal		Parallel		Linear	
	Grad. calls	Energy error	Grad. calls	Energy error	Grad. calls	Energy error
Gradient Grad.	26±4	0.024±0.008	33±9	0.083±0.086	32±8	0.13±0.08
Numerical Grad. ($\delta = 0.05$)	32±8	0.019±0.006	42±13	0.083±0.095	40±11	0.13±0.08
Numerical Grad. ($\delta = 0.1$)	32±8	0.019±0.007	41±13	0.074±0.083	40±12	0.11±0.07

Finally, we explore how control errors influence the optimization with numerical and analytical gradients. Assuming that control errors dominate Δg, we performed 150 runs of the VQE optimization under the influence of control errors, with $\Delta\Theta = 0.01$. Our results, summarized in Table 2.3, compare the average error in the final energy and average number of gradient calls for carrying out the optimization. We observe that the analytical and the numerical gradients provide accuracies in similar ranges. However, the optimization with analytical gradients requires 20% less gradient evaluations in average compared to optimization with numerical gradients. These results suggest that the analytical gradient might have a better convergence under the influence of control errors, in addition to a much lower sampling cost.

2.5 Discussion

We have presented a series of strategies for the calculation of molecular energies using the VQE algorithm combined with a UCC ansatz for carrying out the state preparation. The UCC method provides a hierarchy of wavefunction ansatze that can be prepared using quantum circuits with a size that scales polynomially in the number of orbitals and particles of the system. In particular, the approximation up to double cluster operators provides a good compromise between cost and accuracy for applications in quantum chemistry, with a number of parameters that scales as $O(N^2\eta^2)$. The number of parameters in the approximation determines the size of the circuit and impacts the cost of the classical optimization required for wavefunction optimization.

Additionally, we have illustrated how efficient classical approximations to the amplitudes of the cluster operators, such as those obtained from perturbation theory, can be used to reduce the cost of the VQE algorithm for chemistry. In particular, we showed that classical amplitudes provide effective initial guesses for the optimization process and serve as a pre-screening mechanism to remove cluster operators that have negligible contribution to the optimal wavefunction. This

strategy is an example of a hybrid quantum-classical scheme for quantum simulation, where efficient classical approximations are employed to reduce quantum resources and boost the efficiency of the quantum subroutine. These hybrid schemes are more likely to be the first quantum algorithms to exploit the power of small quantum computers for quantum simulation.

Our numerical analysis also highlights the deficiencies of some derivative-free methods, such as Nelder-Mead and Powell, that have been previously employed in numerical and experimental demonstrations of VQE[255,353]. These methods performs poorly for a relatively large number of parameters, failing to converge to the correct wavefunctions unless a physically meaningful initial guess is employed. Among the methods tested, COBYLA showed a much better performance.

Finally, we introduced an analytical approach to compute the energy gradient for variational circuits and evaluated its performance for the UCC ansatz. This approach allows us to employ gradient based methods to minimize the energy. Our numerical simulations show that our analytical approach provides solutions of the same quality obtained with derivative-free and numerical gradient approaches. In addition, analytical gradients have a much smaller sampling cost than numerical gradients as well as better convergence behavior under the effect of control noise. We point out that our formulation of the analytical gradient can be adapted to other algorithms that employ a quantum-classical hybrid scheme such as the quantum approximate optimization algorithm[104] and some methods proposed in the context of quantum machine learning[20,280,350]. Future work will be devoted to evaluating the performance of gradient-based and gradient-free optimizations under non-coherent errors and state preparation and measurement (SPAM) errors.

2.6 APPENDIX

2.6.1 COMMUTATIVITY OF SUBTERMS IN EXCITATION OPERATORS

Assuming real cluster amplitudes, the JW transformation of the single and double cluster operators for UCC can be written as follows:

$$t_i^a(a_a^\dagger a_i - h.c.) \equiv \frac{it_i^a}{2} \bigotimes_{k=i+1}^{a-1} \sigma_k^z(\sigma_i^y\sigma_a^x - \sigma_i^x\sigma_a^y) \tag{2.52}$$

$$\begin{aligned} t_{ij}^{ab}(a_b^\dagger a_a^\dagger a_j a_i - h.c.) \equiv \frac{it_{ij}^{ab}}{8} \bigotimes_{k=i+1}^{j-1} \sigma_k^z \bigotimes_{l=a+1}^{b-1} \sigma_l^z \\ (\sigma_i^x\sigma_j^x\sigma_a^y\sigma_b^x + \sigma_i^y\sigma_j^x\sigma_a^y\sigma_b^y \\ +\sigma_i^x\sigma_j^y\sigma_a^y\sigma_b^y + \sigma_i^x\sigma_j^x\sigma_a^x\sigma_b^y \\ -\sigma_i^y\sigma_j^x\sigma_a^x\sigma_b^x - \sigma_i^x\sigma_j^y\sigma_a^x\sigma_b^x \\ -\sigma_i^y\sigma_j^y\sigma_a^y\sigma_b^x - \sigma_i^y\sigma_j^y\sigma_a^x\sigma_b^y), \end{aligned} \tag{2.53}$$

where we assume without lost of generality that $b > a > j > i$. The commutativity among the terms in Eq. (2.52) and Eq. (2.53) can be verified by inspection. In general, the JW transformation of an UCC operator of order n will comprise 2^{2n-1} terms, composed by the same string of Z operators multiplying the sum of all the possible strings of X and Y operators acting on $2n$ qubits, such as the numbers of X and Y operators are both odd. The commutativity between any of this terms reduces to the commutativity of the strings containing x and y operators only.

Consider two arbitrary strings of x and y operators of length $2n$, $P_A = \bigotimes_{i=1}^{2n} \sigma_i^{a_i}$ and

$P_B = \bigotimes_{i=1}^{2n} \sigma_i^{b_i}$, acting on the same set of qubits. The commutator is given by:

$$[P_A, P_B] = \bigotimes_{i=1}^{2n} (\sigma_i^{a_i} \sigma_i^{b_i}) - \bigotimes_{i=1}^{2n} (\sigma_i^{b_i} \sigma_i^{a_i}) \tag{2.54}$$

where the product $\sigma_i^{a_i} \sigma_i^{b_i}$ can take three values:

$$\sigma_i^{a_i} \sigma_i^{b_i} = \begin{cases} \mathbf{1} \quad if \quad a_i = b_i \\ i\sigma_z \quad if \quad a_i = x \quad b_i = y \\ -i\sigma_z \quad if \quad a_i = y \quad b_i = x. \end{cases} \tag{2.55}$$

Applying Eq. (2.55) to Eq. (2.54), we can write:

$$[P_A, P_B] = \left[(-i)^{n_y^A - c_y} (i)^{n_x^A - c_x} - (-i)^{n_y^B - c_y} (i)^{n_x^B - c_x} \right] P,$$

where P is the string of Pauli matrices obtained from the multiplication and n_x^A and n_y^A are the numbers of X and Y operators in string A, respectively. n_x^B and n_y^B are defined analogously. c_x is the number of times $a_i = b_i = x$; c_y is defined accordingly. Rearranging the previous equation, we obtain:

$$[P_A, P_B] = (-1)^{n_y^A - c_y} \left(1 - (-1)^{n_y^B - n_y^A} \right) i^{2n - c_x - c_y} P \tag{2.56}$$

Now recall that for the UCC operators, n_y^A and n_y^B are both odd and thus $n_y^B - n_y^A$ is even. Consequently, $[P_A, P_B]$ is zero. We conclude that the subterms comprising a single UCC operator commute.

2.6.2 MAPPING FERMIONIC OPERATORS TO QUBIT OPERATORS

In order to implement the a fermionic Hamiltonian on a quantum computer or prepare a wavefunction according to a given ansatz expressed in terms of fermionic second quantized operators, we need to map the algebra of fermionic operators to qubit operations. The occupation basis employed in second quantization can be mapped directly to a qubit basis, where the same space is represented by the space spanned by the tensor product of N qubits[300]:

$$|f_{N-1}, \cdots, f_0\rangle \rightarrow |q_{N-1}\rangle \otimes \cdots \otimes |q_0\rangle \tag{2.57}$$

In the qubit occupation basis, a natural choice of second quantized operators would be the qubit creator and annihilation operator defined as: $\sigma_j^+ \equiv \frac{1}{2}(\sigma_j^x - i\sigma_j^y)$ and $\sigma_j^- \equiv \frac{1}{2}(\sigma_j^x + i\sigma_j^y)$, respectively. However, these operators do not obey the fermionic anticommutation relations and therefore cannot be employed to map the fermionic creation and annihilation operators. To account for the anticommutation properties, we employ the Jordan-Wigner (JW) transformation instead[165], where the fermionic second quantized operators are mapped according to the rules:

$$a_j^\dagger \equiv \frac{1}{2}(\sigma_j^x - i\sigma_j^y) \otimes \sigma_{j-1}^z \otimes \cdot \otimes \sigma_0^z \tag{2.58}$$

$$a_j \equiv \frac{1}{2}(\sigma_j^x + i\sigma_j^y) \otimes \sigma_{j-1}^z \otimes \cdot \otimes \sigma_0^z \tag{2.59}$$

The effect of the string of σ_z gates is to introduce the correct phase associated with the application of fermionic operators, which depends on the parity of the set of qubits with index less than j. In the JW mapping the occupancy of the orbitals is stored locally in every qubit while the parity is a non-local property that needs to be determined by performing σ_z operations on a subset of the qubits. Consequently, this mapping generates local terms with length $O(N)$. An alternative mapping known as the Bravyi-Kitaev (BK) transformation, use an intermediate approach, where

some qubits store information in the occupation number representation while others store information in the parity representation, achieving scaling $O(\log N)$ in the length of the local terms in the Hamiltonian. The details about the derivation of the BK transformation can be found in[300,336]. Using either the JW transformation or the BK transformation the quantum chemistry Hamiltonian can be transformed to a sum of local qubit terms, expressed as strings of Pauli matrices.

3

A low-depth circuit ansatz for variational quantum simulation of fermionic systems

Apart from minor modifications, this chapter originally appeared as[279]:

"Low-depth circuit ansatz for preparing correlated fermionic states on a quantum computer". Pierre-Luc Dallaire-Demers, Jonathan Romero, Libor Veis, Sukin Sim and Alán Aspuru-Guzik. arXiv:1801.01053 (2018). Reproduced with permission of the authors.

3.1 Introduction

The macroscopic properties of matter emerge from its microscopic quantum constituents whose massive components are mostly fermions. Understanding and modeling the behavior of a large number of interacting fermions is a central and fundamental problem in Physics and Chemistry which requires a large investment in computational resources as the memory required to represent a many-body state scales exponentially with the number of particles. Therefore, a computer operating on quantum mechanical principles have the potential to revolutionize the simulation of quantum systems[4,108]. Such a machine would improve our ability to design new molecules such as drugs and catalysts[272], build new superconducting[27,79,355] and topological materials and improve our understanding of nuclear matter. Algorithm leveraging the advantages of quantum computers for quantum simulations have steadily been developed in the past two decades[14,16,24,27,143,170,171,194,248,249,255,265,341,355,359,374] as quantum processors are scaling in

size[206,237,241]. Variational quantum eigensolvers (VQE) have recently appeared as a promising class of quantum algorithms designed to prepare states for quantum simulations[226,248,353]. However, near-term devices will suffer from limited coherence as a consequence of noise and finite experimental precision[202,331]. This incentives the search for low-depth circuits for quantum simulations and state preparation[19,168].

In this chapter, we present a new type of low-depth VQE ansatz motivated by the Bogoliubov coupled cluster theory[277,310,325]. Our approach can be used to prepare the ground state of correlated fermions with pairing interactions by systematically appending variational cycles composed of linear-depth blocks of 2-qubit gates. In section 3.2, we first review the formulation of the strong correlation problem for fermions in the context of second quantization. We then present the unitary version of Bogoliubov coupled cluster theory and review how the generalized Hartree-Fock (GHF) reference state can be computed as a fermionic Gaussian state. Using the theory of matchgates, we show how pure fermionic Gaussian states can be exactly prepared on a quantum computer using a linear-depth circuit. Finally, we introduce the low-depth circuit ansatz (LDCA), consisting of the previous matchgate circuit plus additional nearest-neighbor phase coupling. We numerically benchmark the LDCA in section 3.3 for the prototypical examples of the Fermi-Hubbard model in condensed matter and the automerization reaction of cyclobutadiene in quantum chemistry, showing its potential to describe the exact ground state of strongly correlated systems.

3.2 Generalized variational quantum eigensolver

In this section, we review and extend the theoretical foundations of VQE. Specifically, in subsection 3.2.1, we review the definition of finding the ground state of fermionic Hamiltonians as found in quantum chemistry, condensed matter, and nuclear physics. In subsection 3.2.2 we introduce the Bogoliubov unitary coupled cluster (BUCC) theory as a variational ansatz to the

ground state problem. In subsection 3.2.3 we review the formalism of the GHF theory as this is the starting point of the BUCC optimization method as well as the new method presented in the following subsection. In subsection 3.2.4 we show how a GHF state can be prepared on a quantum processor using matchgates and introduce a LDCA which can be used to prepare the ground state of fermionic Hamiltonian with surprisingly high accuracy. Finally, in subsection 3.2.5, we outline an implementation to compute the analytical gradient of the LDCA using quantum resources.

3.2.1 Formulation of the problem

Many systems in quantum chemistry[327], condensed matter[197,274,303], and nuclear structure physics[31,275] can be modeled by an ensemble of interacting fermions (electrons, nucleons) described by a second quantized Hamiltonian of the form

$$\begin{aligned} H &= \sum_{pq}\left(t_{pq}a_p^\dagger a_q + \Delta_{pq}a_p^\dagger a_q^\dagger + \Delta_{pq}^* a_q a_p\right) \\ &\quad + \sum_{pqrs} v_{pqrs} a_p^\dagger a_q^\dagger a_s a_r \\ &\quad + \sum_{pqrstu} w_{pqrstu} a_p^\dagger a_q^\dagger a_r^\dagger a_u a_t a_s . \end{aligned} \tag{3.1}$$

In general, the p,q,...,u indices run over all relevant quantum numbers (e.g. position, momentum, band number, spin, angular momentum, isospin, etc) which define M fermionic modes. The fermionic mode operators follow canonical anti-commutation relations $\left\{a_k, a_l^\dagger\right\} = \delta_{kl}$ and $\{a_k, a_l\} = \left\{a_k^\dagger, a_l^\dagger\right\} = 0$. The kinetic energy terms t_{pq} and the interaction v_{pqrs} are ubiquitous in most theories, while pairing terms Δ_{pq} often appear in the context of mean-field superconductivity, and the three-body interaction term w_{pqrstu} can be phenomenologically introduced in nuclear physics[213].

As a prerequisite to calculating various observable quantities, we are interested in finding the ground state $\rho_0 = |\Psi_0\rangle\langle\Psi_0|$ of the Hamiltonian (3.1) such that the energy E is minimized over the

set of all possible states ρ in a given Hilbert space:

$$\begin{aligned} E_0 &\equiv E(\rho_0) \\ &= \min_\rho E(\rho) \\ &= \min_\rho \mathrm{tr}(H\rho). \end{aligned} \tag{3.2}$$

When this minimization cannot be done either analytically or with numerically exact methods, we have to resort to approximate methods such as variational ansatzes. One such ansatz, the BUCC method, is defined in the next subsection.

3.2.2 Bogoliubov unitary coupled cluster theory

Coupled cluster methods are used in ab initio quantum chemistry calculations to describe correlated many-body states with a better accuracy than the Hartree-Fock method. Bogoliubov- and quasiparticle-based coupled cluster methods extends the range of applicability of those methods to systems with mean-field paired states[277,310,325]. Anticipating the implementation on quantum computers, we present the formalism for the unitary version of the Bogoliubov coupled cluster theory. We first review the Bogoliubov transformation and the parametrization of the ansatz.

The most general linear transformation acting on fermionic creation and annihilation operators that preserves the canonical anti-commutation relation is the Bogoliubov transformation. In this transformation, the quasiparticle operators $\left(\beta_{p'}^\dagger;\beta_{p'}\right)$ are related to the single-particle operators $\left(a_p^\dagger;a_p\right)$ by a unitary matrix

$$\begin{aligned} \beta_{p'}^\dagger &= \sum_p \left(U_{pp'}a_p^\dagger + V_{pp'}a_p\right) \\ \beta_{p'} &= \sum_p \left(U_{pp'}^* a_p + V_{pp'}^* a_p^\dagger\right). \end{aligned} \tag{3.3}$$

This transformation preserves the canonical anti-commutation relation such that $\left\{\beta_k, \beta_l^\dagger\right\} = \delta_{kl}$

and $\{\beta_k, \beta_l\} = \left\{\beta_k^\dagger, \beta_l^\dagger\right\} = 0$. By introducing the vector notation $\vec{a}^\top = \left(a_1, \ldots, a_M, a_1^\dagger, \ldots, a_M^\dagger\right)$ and $\vec{\beta}^\top = \left(\beta_1, \ldots, \beta_M, \beta_1^\dagger, \ldots, \beta_M^\dagger\right)$, it is easy to express (3.3) in matrix notation as $\vec{\beta} = \mathcal{U}\vec{a}$ where the Bogoliubov transformation is unitary $\mathcal{U}^{-1} = \mathcal{U}^\dagger$ and its matrix is defined as

$$\mathcal{U} = \begin{pmatrix} \mathbf{U}^* & \mathbf{V}^* \\ \mathbf{V} & \mathbf{U} \end{pmatrix}. \tag{3.4}$$

The ground state of a quadratic Hamiltonian (all $v_{pqrs} = 0$ and $w_{pqrstu} = 0$) is a product state

$$|\Phi_0\rangle = C \prod_{k=1}^{M} \beta_k |\text{vac}\rangle, \tag{3.5}$$

where $|\text{vac}\rangle$ is the Fock vacuum and C is a normalization factor. If the ground state is not degenerate, (3.5) acts as a quasiparticle vacuum $\beta_j |\Phi_0\rangle = 0$.

We can define the quasiparticle cluster operator $\mathcal{T} = \mathcal{T}_1 + \mathcal{T}_2 + \mathcal{T}_3 + \ldots$ where

$$\begin{aligned} \mathcal{T}_1 &= \textstyle\sum_{k_1 k_2} \theta_{k_1 k_2} \beta_{k_1}^\dagger \beta_{k_2}^\dagger \\ \mathcal{T}_2 &= \textstyle\sum_{k_1 k_2 k_3 k_4} \theta_{k_1 k_2 k_3 k_4} \beta_{k_1}^\dagger \beta_{k_2}^\dagger \beta_{k_3}^\dagger \beta_{k_4}^\dagger \\ \mathcal{T}_3 &= \textstyle\sum_{k_1 k_2 k_3 k_4 k_5 k_6} \theta_{k_1 k_2 k_3 k_4 k_5 k_6} \beta_{k_1}^\dagger \beta_{k_2}^\dagger \beta_{k_3}^\dagger \beta_{k_4}^\dagger \beta_{k_5}^\dagger \beta_{k_6}^\dagger. \end{aligned} \tag{3.6}$$

The $\theta_{k_1 k_2 \ldots} \in \mathbb{C}$ are variational parameters which are fully antisymmetric such that $\theta_{k_1 k_2 \ldots} = (-1)^{\xi(P)} \theta_{P(k_1 k_2 \ldots)}$, where $\xi(P)$ is the signature of the permutation P. The BUCC ansatz is defined as

$$|\Psi(\Theta)\rangle = e^{i\left(\mathcal{T}(\Theta) + \mathcal{T}^\dagger(\Theta)\right)} |\Phi_0\rangle. \tag{3.7}$$

where Θ corresponds to the set of variational parameters $\theta_{k_1 k_2 \ldots}$ and $|\Phi_0\rangle$ is a reference state. Since the transformation is unitary $|\langle \Psi(\Theta) | \Psi(\Theta) \rangle| = 1$, $|\Psi(\Theta)\rangle$ is always normalized. The BUCC ansatz is said to be over single (BUCCS) or double excitations (BUCCSD) if the cluster operator $\mathcal{T}$

is truncated at the first or second order.

To variationally optimize the BUCC ansatz, we aim to find the angles Θ that minimize the energy

$$\min_{\Theta} E(\Theta) = \langle \Psi(\Theta) | H | \Psi(\Theta) \rangle \tag{3.8}$$

subject to the constraint that the number of particles

$$\begin{aligned} \langle N(\Theta) \rangle &= \langle \Psi(\Theta) | N | \Psi(\Theta) \rangle \\ &= \sum_{p=1}^{M} \langle \Psi(\Theta) | a_p^\dagger a_p | \Psi(\Theta) \rangle \end{aligned} \tag{3.9}$$

should be kept constant, as the quasiparticles operators generally do not preserve the total particle number. In the next subsection we will explicitly show how to compute the reference state from the generalized Hartree-Fock theory before describing the details of the implementation of the quantum algorithm.

3.2.3 Generalized Hartree-Fock theory

Here we show how to obtain the Bogoliubov matrix (3.4) used to define the reference state (3.5). The method relies on the theory of fermionic Gaussian states [187,188] for which we review the formalism and a method to obtain the covariance matrix of the ground state without a self-consistent loop. Fermionic Gaussian states are a useful starting point for quantum simulations as they include the family of Slater determinants from Hartree-Fock theory and Bardeen-Cooper-Schrieffer (BCS) states found in the mean-field theory of superconductivity [21,73] and can be easily prepared on a quantum computer [344].

For M fermionic modes, it is convenient to define the $2M$ Majorana operators

$$\begin{aligned} \gamma_j &= \gamma_j^A &&= a_j^\dagger + a_j \\ \gamma_{j+M} &= \gamma_j^B &&= -i\left(a_j^\dagger - a_j\right) \end{aligned} \tag{3.10}$$

as the fermionic analogues of position and momentum operators. Let's note that we used either the extended index notation (from 1 to $2M$) or the A,B superscript notation interchangeably throughout the chapter to make the equations clearer. Their commutation relation satisfies $\{\gamma_k, \gamma_l\} = 2\delta_{kl}$ such that $\gamma_k^2 = 1$. It is useful to define the vector notation $\vec{\gamma}^\top = (\gamma_1, \ldots, \gamma_M, \gamma_{M+1}, \ldots, \gamma_{2M})$ and write $\vec{\gamma} = \Omega\vec{a}$ where

$$\Omega = \begin{pmatrix} \mathbf{1} & \mathbf{1} \\ i\mathbf{1} & -i\mathbf{1} \end{pmatrix}. \tag{3.11}$$

In this case, $\mathbf{1}$ is the $M \times M$ identity matrix. A general fermionic Gaussian state[188] has the form of the exponential of a quadratic product of fermionic operators

$$\rho = \frac{1}{Z} e^{-\frac{i}{4}\vec{\gamma}^\top G \vec{\gamma}}, \tag{3.12}$$

where Z is the normalization factor and G is a real and antisymmetric matrix such that $G^\top = -G$. It can be fully characterized by a real and antisymmetric covariance matrix which is defined by

$$\Gamma_{kl} = \frac{i}{2}\mathrm{tr}\left(\rho\left[\gamma_k, \gamma_l\right]\right), \tag{3.13}$$

where $[\cdot,\cdot]$ is the commutator. For a pure Gaussian state, $\Gamma^2 = -\mathbf{1}$, where $\mathbf{1}$ is the $2M \times 2M$ identity matrix. In general, the purity is given by $\chi = -\frac{1}{2M}\mathrm{tr}\left(\Gamma^2\right)$. In order to extract $\mathcal{U}$ given a

covariance matrix Γ, we make use of the complex covariance matrix representation

$$\Gamma_c = \frac{1}{4}\Omega^\dagger \Gamma \Omega^* = \begin{pmatrix} \mathbf{Q} & \mathbf{R} \\ \mathbf{R}^* & \mathbf{Q}^* \end{pmatrix}, \tag{3.14}$$

where $\mathbf{Q}_{kl} = \frac{i}{2}\langle [a_k, a_l] \rangle$ and $\mathbf{R}_{kl} = \frac{i}{2}\left\langle \left[a_k, a_l^\dagger\right] \right\rangle$ (expectation values are defined as $\langle O \rangle = \mathrm{tr}\,(O\rho)$). From there, we can define the single-particle density operators $\kappa \equiv -i\mathbf{Q}$ and $\rho \equiv \frac{1}{2}\mathbf{1} - i\mathbf{R}^\top$ and recast the Gaussian state in the form of a single-particle density matrix

$$\mathcal{M} = \begin{pmatrix} \rho & \kappa^\dagger \\ \kappa & \mathbf{1} - \rho^\top \end{pmatrix} \tag{3.15}$$

such that $\mathcal{M}^2 = \mathcal{M}$ for pure states[42]. If we define the matrix $\mathcal{E} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{pmatrix}$, then it is possible to find the Bogoliubov transformation (3.4) with the eigenvalue equation

$$\mathcal{M}\mathcal{U}^\dagger = \mathcal{E}\mathcal{U}^\dagger. \tag{3.16}$$

Next, we show how to compute the covariance matrix (3.13) approximating the ground state of the Hamiltonian (3.1).

3.2.3.1 FINDING THE GROUND STATE

These steps are a review of the method found in[187] aimed at calculating the covariance matrix approximating the ground state of an interacting Hamiltonian without a self-consistent loop.

The Hamiltonian (3.1) can be rewritten with Majorana operators in the form

$$\begin{aligned} H &= i\sum_{pq} T_{pq}\gamma_p\gamma_q \\ &\quad +\sum_{pqrs} V_{pqrs}\gamma_p\gamma_q\gamma_s\gamma_r \\ &\quad +i\sum_{pqrstu} W_{pqrstu}\gamma_p\gamma_q\gamma_r\gamma_u\gamma_t\gamma_s, \end{aligned} \tag{3.17}$$

where $T^\top = -T$ and V and W are antisymmetric under the exchange of any two adjacent indices. Expectation values over gaussian states can be efficiently calculated using Wick's theorem which has the form

$$i^p \mathrm{tr}\left(\rho\gamma_{j_1}\ldots\gamma_{j_{2p}}\right) = \mathrm{Pf}\left(\Gamma|_{j_1\ldots j_{2p}}\right), \tag{3.18}$$

where $1 \leq j_1 < \ldots < j_{2p} \leq 2M$, $\Gamma|_{j_1\ldots j_{2p}}$ is the corresponding submatrix of Γ and

$$\begin{aligned} \mathrm{Pf}(\Gamma) &= \tfrac{1}{2^M M!}\sum_{s\in S_{2M}} \mathrm{sgn}(s)\prod_{j=1}^{M}\Gamma_{s(2j-1),s(2j)} \\ &= \sqrt{\det(\Gamma)} \end{aligned} \tag{3.19}$$

is the Pfaffian of a $2M\times 2M$ matrix defined from the symmetric group S_{2M} where $\mathrm{sgn}(s)$ is the signature of the permutation s. Assuming that Wick's theorem holds, we can write an effective but state dependent quadratic Hamiltonian

$$h(\Gamma) = T + 6\mathrm{tr}_B(V\Gamma) + 45\mathrm{tr}_C(W\Gamma\Gamma), \tag{3.20}$$

where $\mathrm{tr}_B(V\Gamma)_{ij} = \sum_{kl} V_{ijkl}\Gamma_{lk}$ and $\mathrm{tr}_C(W\Gamma\Gamma)_{ij} = \sum_{klmn} W_{ijklmn}\Gamma_{kn}\Gamma_{ml}$. To get the covariance matrix of the reference state, we use the imaginary time evolution starting from a pure state $\Gamma(0)^2 = -\mathbf{1}$:

$$\Gamma(\tau) = O(\tau)\Gamma(0)O(\tau)^\top, \tag{3.21}$$

where the orthogonal time evolution operator is given by

$$O(\tau) = \mathbb{T} e^{2\int_0^\tau d\tau' [h(\Gamma(\tau')),\Gamma(\tau')]}, \tag{3.22}$$

with $\mathbb{T}$ being the time ordering. The steady state is reached when $[h(\Gamma),\Gamma] = 0$. This is guaranteed to lower the energy of an initial state and keep the purity of the initial $\Gamma(0)$ but the imaginary time evolution may get stuck in a local minimum. A second complementary approach consists in minimizing the free energy of (3.17). The procedure simply involves fixed point iterations on the transcendental equation

$$\Gamma = \lim_{\beta\to\infty} \tanh\left[2i\beta h(\Gamma)\right]. \tag{3.23}$$

In our numerical experiments, we find that an imaginary time evolution (3.21) followed by a fixed point evolution (3.23) is numerically stable and consistently reaches the desired GHF ground state. In the following subsection, we will show how the theory of matchgates can be used to prepare a pure Gaussian state on a quantum computer as a reference state for a variational procedure.

3.2.4 The quantum subroutine

It is expected that quantum computer will enable the simulation of quantum systems beyond the reach of classical computers. An important challenge for practical simulations is to prepare the ground state of interesting Hamiltonians with high accuracy. The VQE protocol[168,226,248,255,353] suggests a general procedure to reach this ground state. However, current implementations of the protocol have to trade long circuit depth for accuracy in a non-controllable manner. In this subsection, we introduce a composable VQE ansatz which is both accurate and hardware efficient with the added advantage of being able to represent states with BCS-like pairing correlations. Our method relies on the theory of matchgates and its relation to fermionic linear optics[48,49,167,333,339,344] to both prepare a reference Gaussian state and parametrize an ansatz with a

transformation analogous to fermionic non-linear optics. After a brief review of the theory of matchgates, we show how a given pure Gaussian state can be prepared on a quantum register with a linear-depth algorithm. A different algorithm with the same scaling was recently proposed in [160]. Unlike the procedure in [160] that relies on a gate decomposition strategy, our method has a fixed circuit structure with variable parameters. We then proceed to introduce a low-depth circuit ansatz with inherited properties of the BUCC ansatz and the apparent accuracy of the full configuration interaction method.

3.2.4.1 MATCHGATE DECOMPOSITION OF A BOGOLIUBOV TRANSFORMATION

In the computational basis of a 2-qubit Hilbert space, matchgates [339] have the general form

$$\mathcal{G}(A,B) = \begin{pmatrix} p & 0 & 0 & q \\ 0 & w & x & 0 \\ 0 & y & z & 0 \\ r & 0 & 0 & s \end{pmatrix}, \tag{3.24}$$

where $A = \begin{pmatrix} p & q \\ r & s \end{pmatrix}$ and $B = \begin{pmatrix} w & x \\ y & z \end{pmatrix}$ are $SU(2)$ matrices with the same determinant $\det A = \det B$. They form a group which is generated by the tensor product of nearest-neighbor Pauli operators

$$\begin{aligned} \sigma_x^j \otimes \sigma_x^{j+1} &= -i\gamma_j^B \gamma_{j+1}^A \\ \sigma_x^j \otimes \sigma_y^{j+1} &= -i\gamma_j^B \gamma_{j+1}^B \\ \sigma_y^j \otimes \sigma_x^{j+1} &= i\gamma_j^A \gamma_{j+1}^A \\ \sigma_y^j \otimes \sigma_y^{j+1} &= i\gamma_j^A \gamma_{j+1}^B \\ \sigma_z^j \otimes \mathbb{I}^{j+1} &= -i\gamma_j^A \gamma_j^B \\ \mathbb{I}^j \otimes \sigma_z^{j+1} &= -i\gamma_{j+1}^A \gamma_{j+1}^B, \end{aligned} \tag{3.25}$$

which also correspond to the Jordan-Wigner transformed product of all products of nearest-neighbor Majorana operators, therefore establishing the connection with fermionic gaussian operations. The Bogoliubov transformation (3.3) can be written as an $SO(2M)$ transformation of the Majorana operators (3.10) as $\vec{\gamma}' = \mathcal{R}\vec{\gamma}$, where

$$\mathcal{R} = \begin{pmatrix} \mathrm{Re}(\mathbf{U}+\mathbf{V}) & -\mathrm{Im}(\mathbf{U}-\mathbf{V}) \\ \mathrm{Im}(\mathbf{U}+\mathbf{V}) & \mathrm{Re}(\mathbf{U}-\mathbf{V}) \end{pmatrix}. \tag{3.26}$$

To implement this transformation on a quantum processor, there exists a quantum circuit of nearest-neighbor matchgates U_{Bog} acting on M qubits [167] such that

$$U_{\mathrm{Bog}}\gamma_j U_{\mathrm{Bog}}^{\dagger} = \sum_{k=1}^{2M} \mathcal{R}_{kj}\gamma_k. \tag{3.27}$$

An example of such a circuit known as the fermionic fast Fourier transform is described in [344]. In general, the Hoffman algorithm [149] can be used to decompose U_{Bog} in $2M(M-1)$ $SO(4)$ rotations between pairs of modes and M $SO(2)$ local phases. In total, these $2M^2 - M$ angles correspond to the same number of quantum gates. Using the fact that quantum gates can be operated in parallel in a linear chain of qubits, any transformation $\mathcal{R}$ can be implemented in circuit depth $8\left\lceil \frac{M}{2} \right\rceil + 1$, as detailed in Figure 3.1.

Since the Hoffman method assumes sequential operations on each pair of modes, we used an optimal control scheme [177,215] in $SO(2M)$ to allow an easy parametrization of gates acting in parallel. This is generally efficient on a classical computer since the matchgates only operate on a much smaller subspace of the full $SU\left(2^M\right)$ transformation allowed on M qubits. The

transformation $\mathcal{R}$ can be decomposed in local and nearest-neighbor mode rotations such that

$$\begin{aligned}\mathcal{R} &= \prod_{k=1}^{\lceil \frac{M}{2} \rceil} \left\{ \prod_{\mu,\nu} \prod_{j \in \text{odd}} r_{j,j+1}^{\mu\nu} \left(\theta_{j,j+1}^{\mu\nu(k)} \right) \right. \\ &\quad \left. \times \prod_{\mu,\nu} \prod_{j \in even} r_{j,j+1}^{\mu\nu} \left(\theta_{j,j+1}^{\mu\nu(k)} \right) \right\} \\ &\quad \times \prod_{j=1}^{M} r_{jj}^{AB} \left(\theta_{jj}^{AB} \right),\end{aligned} \tag{3.28}$$

where $\mu, \nu \in \{A, B\}$ and $j \in \{1, \ldots, M\}$. The mode rotations are parametrized by the $2M^2 - M$ angles $\theta_{ij}^{\mu\nu(k)}$

$$r_{ij}^{\mu\nu} = e^{2\theta_{ij}^{\mu\nu} h_{ij}^{\mu\nu}} \tag{3.29}$$

with $SO(2M)$ Hamiltonians

$$h_{ij}^{\mu\nu} = \delta_{i\mu, j\nu} - \delta_{j\nu, i\mu}. \tag{3.30}$$

The optimal control method maximizes the fidelity function

$$\Phi = \frac{1}{2M} \text{tr} \left\{ \mathcal{R}_{\text{target}}^{\top} \mathcal{R}(\Theta) \right\} \tag{3.31}$$

using the gradient

$$\frac{\partial r_{ij}^{\mu\nu}}{\partial \theta_{kl}^{\alpha\beta}} = 2 h_{ij}^{\mu\nu} r_{ij}^{\mu\nu} \delta_{\alpha\mu} \delta_{\beta\nu} \delta_{ki} \delta_{lj}. \tag{3.32}$$

As shown in Figure 3.1 on a 8-qubit example, this decomposition explicitly translates into a quantum circuit of single qubit phase-rotations

$$R_j^Z = e^{i\theta_{ii}^{AB} \sigma_z^i} \tag{3.33}$$

and nearest-neighbor matchgates

$$G_{ij}^{(k)} = R_{ij}^{XX(k)} R_{ij}^{-YY(k)} R_{ij}^{XY(k)} R_{ij}^{-YX(k)}, \tag{3.34}$$

where each rotation corresponds to

$$\begin{aligned} R_{ij}^{-YX(k)} &= e^{-i\theta_{ij}^{AA(k)} \sigma_y^i \otimes \sigma_x^j} \\ R_{ij}^{XY(k)} &= e^{i\theta_{ij}^{BB(k)} \sigma_x^i \otimes \sigma_y^j} \\ R_{ij}^{-YY(k)} &= e^{-i\theta_{ij}^{AB(k)} \sigma_y^i \otimes \sigma_y^j} \\ R_{ij}^{XX(k)} &= e^{i\theta_{ij}^{BA(k)} \sigma_x^i \otimes \sigma_x^j}. \end{aligned} \tag{3.35}$$

Each parallel cycle interleaves gates between even and odd nearest neighbors

$$U_{\mathrm{MG}}^{(k)} = \prod_{i \in \mathrm{odd}} G_{i,i+1}^{(k)} \prod_{i \in \mathrm{even}} G_{i,i+1}^{(k)} \tag{3.36}$$

and there are $\lceil \frac{M}{2} \rceil$ cycles in total:

$$U_{\mathrm{MG}}^{\mathrm{NN}} = \prod_{k=1}^{\lceil \frac{M}{2} \rceil} U_{\mathrm{MG}}^{(k)}. \tag{3.37}$$

Finally, the unitary Bogoliubov transformation can be composed as

$$U_{\mathrm{Bog}} = U_{\mathrm{MG}}^{\mathrm{NN}} \prod_{i=1}^{M} R_i^Z \tag{3.38}$$

and is also a gaussian operation of the form $U_{\mathrm{Bog}} = e^{i\sum_{pq} \tau_{pq} \gamma_p \gamma_q}$, where $\tau^\top = -\tau$. In the case where the reference state is a Slater determinant, only number-conserving matchgates are required to prepare the state and the depth of the circuit would scale as $4\lceil \frac{M}{2} \rceil + 1$ (since all $\theta_{ij}^{AA(k)}$ and $\theta_{ij}^{BB(k)}$ are set to zero). It should be noticed that a unitary coupled cluster ansatz truncated at first order $e^{i\left(\mathcal{T}_1(\Theta) + \mathcal{T}_1^\dagger(\Theta)\right)}$

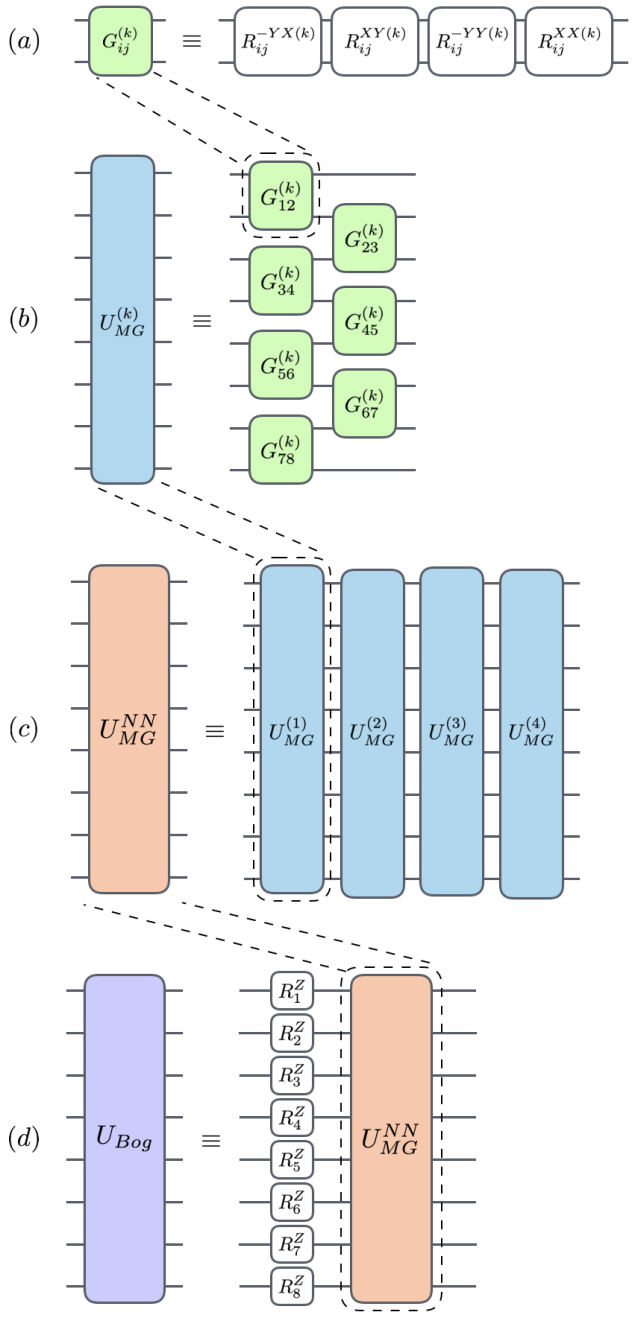


Figure 3.1: Example on 8 qubits of the decomposition of U_{Bog} in a circuit of local phase rotations and nearest-neighbor matchgates. In (a), each $G_{ij}^{(k)}$ is a 2-local operation between qubits i and j composed of 4 rotations for a layer k. As shown in (b), the unitary $U_{\text{MG}}^{(k)}$ for each layer k is built by operating $G_{ij}^{(k)}$'s in parallel first on the even pairs of qubits and then on the odd pairs. Then in (c), the complete sequence of nearest-neighbor matchgates $U_{\text{MG}}^{\text{NN}}$ is composed by a sequence of $\lceil \frac{M}{2} \rceil$ layers. In (d), single qubit phase rotations R_j^Z are used to complete the U_{Bog} circuit.

is also a gaussian transformation and can be implemented in the same way as U_{Bog} with no

trotterization. In what follows, we introduce a VQE scheme that builds on this observation by introducing non-matchgate variational terms into a gate sequence similar to the U_{Bog} decomposition.

3.2.4.2 A LOW-DEPTH CIRCUIT ANSATZ

The Bogoliubov transformation (3.38) acts as a change of basis of the fermionic modes. Therefore, one can simply follow the VQE protocol[255] to implement the BUCC ansatz (3.7) and measure the expectation values $\langle \tilde{H} \rangle = \left\langle U_{\mathrm{Bog}} H U_{\mathrm{Bog}}^{\dagger} \right\rangle$ and $\langle \tilde{N} \rangle = \left\langle U_{\mathrm{Bog}} N U_{\mathrm{Bog}}^{\dagger} \right\rangle$ in the modified basis to prepare an approximate ground state of (3.1). This has the advantage of extending the range of Hamiltonians that can be processed to those with non-number conserving terms (like pairing fields) when compared to the traditional unitary coupled cluster ansatz. However, the change of basis may significantly increase the number of terms that have to be measured. In order to reduce the number of measurements in the VQE protocol, one can start in the product state (3.5) and carry out the variational unitary (3.7) in the quasiparticle basis, followed by an inverse Bogoliubov transformation using matchgates and measurement of the expectation values of the Hamiltonian (3.1) and the number operator N in the original fermionic orbital basis. In the quasiparticle basis, we can map the Bogoliubov operators to qubit operators with the Jordan-Wigner transformation[164,249,300] since they follow the canonical anti-commutation relation

$$\begin{aligned} \beta_p^{\dagger} &= (-1)^{p-1} \left(\bigotimes_{j=1}^{p-1} \sigma_z \right) \otimes \sigma_+ \\ \beta_p &= (-1)^{p-1} \left(\bigotimes_{j=1}^{p-1} \sigma_z \right) \otimes \sigma_- \end{aligned} \tag{3.39}$$

and use the same mapping for Fermionic operators $a_p^{\dagger}$ and a_p after the Bogoliubov transformation. Still, assuming that the number of fermionic particles is proportional to the number of orbitals, a major caveat of BUCCSD-like schemes is that the number of variational parameters will scale as

$O\left(M^4\right)$. In the Jordan-Wigner picture, these terms can be implemented with $O\left(M^6\right)$ gates[137,279]. It is expected that near-term quantum processor will continue to suffer from error rates that make this type of scaling impractical, and therefore more hardware-efficient VQE schemes must be sought[168].

Given that the gate decomposition of U_{Bog} can also exactly parametrize a BUCCS VQE protocol in linear circuit depth, we propose using a scheme augmented with nearest-neighbor phase coupling $\sigma_z \otimes \sigma_z$ rotations to mimic the effects of the quartic variational terms of $\mathcal{T}_2$. Related ideas have already been explored in efficient classical non-gaussian variational methods with great success[307]. In a loose sense, our scheme is a parametrized fermionic non-linear optics circuit that does not involve any trotterization of the variational terms. The algorithm is illustrated in Figure 3.2. As a first step, the quasiparticle vacuum (3.5) is prepared in the Bogoliubov picture with $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ gates acting on each qubits to yield the state $|1\rangle^{\otimes M}$ in the computational basis. In what follows, we will define a L-cycle ansatz built from nearest-neighbor variational matchgates augmented with $\sigma_z \otimes \sigma_z$ rotations. The measurement of the expectation values can be done in the original basis by applying the inverse Bogoliubov transformation $U_{\mathrm{Bog}}^{\dagger}$ defined previously.

In a cycle l of the low-depth circuit ansatz (LDCA), the nearest-neighbor matchgates (3.34) are replaced by

$$\begin{aligned} K_{ij}^{(k,l)}\left(\Theta_{i,j}^{(k,l)}\right) &= R_{ij}^{XX(k,l)} R_{ij}^{-YY(k,l)} \\ &\times R_{ij}^{ZZ(k,l)} R_{ij}^{XY(k,l)} R_{ij}^{-YX(k,l)}, \end{aligned} \tag{3.40}$$

where the rotations are defined as

$$\begin{aligned} R_{ij}^{-YX(k,l)} &= e^{-i\theta_{ij}^{-YX(k,l)}\sigma_y^i\otimes\sigma_x^j} \\ R_{ij}^{XY(k,l)} &= e^{i\theta_{ij}^{XY(k,l)}\sigma_x^i\otimes\sigma_y^j} \\ R_{ij}^{ZZ(k,l)} &= e^{i\theta_{ij}^{ZZ(k,l)}\sigma_z^i\otimes\sigma_z^j} \\ R_{ij}^{-YY(k,l)} &= e^{-i\theta_{ij}^{-YY(k,l)}\sigma_y^i\otimes\sigma_y^j} \\ R_{ij}^{XX(k,l)} &= e^{i\theta_{ij}^{XX(k,l)}\sigma_x^i\otimes\sigma_x^j}. \end{aligned} \tag{3.41}$$

Each layer k applies those variational rotations in parallel first on the even pairs and then on the odd pairs such that

$$\begin{aligned} U_{\text{VarMG}}^{(k,l)}\left(\Theta^{(k,l)}\right) &= \prod_{i\in\text{odd}} K_{i,i+1}^{(k,l)}\left(\Theta_{i,i+1}^{(k,l)}\right) \\ &\quad \times \prod_{i\in\text{even}} K_{i,i+1}^{(k,l)}\left(\Theta_{i,i+1}^{(k,l)}\right). \end{aligned} \tag{3.42}$$

A cycle l is composed of $\left\lceil \frac{M}{2} \right\rceil$ layers such that the variational ansatz is equivalent to a BUCCS transformation when the $\theta_{ij}^{ZZ(k,l)}$ are equal to zero:

$$U_{\text{VarMG}}^{\text{NN}(l)}\left(\Theta^{(l)}\right) = \prod_{k=1}^{\left\lceil \frac{M}{2} \right\rceil} U_{\text{VarMG}}^{(k,l)}\left(\Theta^{(k,l)}\right). \tag{3.43}$$

Finally, the L cycle are assembled sequentially to form the complete variational ansatz

$$U_{\text{VarMG}}\left(\Theta\right) = \prod_{l=1}^{L} U_{\text{VarMG}}^{\text{NN}(l)}\left(\Theta^{(l)}\right) \prod_{i=1}^{M} R_i^Z\left(\theta_i^Z\right), \tag{3.44}$$

with only one round of variational phase rotations

$$R_i^Z\left(\theta_i^Z\right) = e^{i\theta_i^Z\sigma_z^i}. \tag{3.45}$$

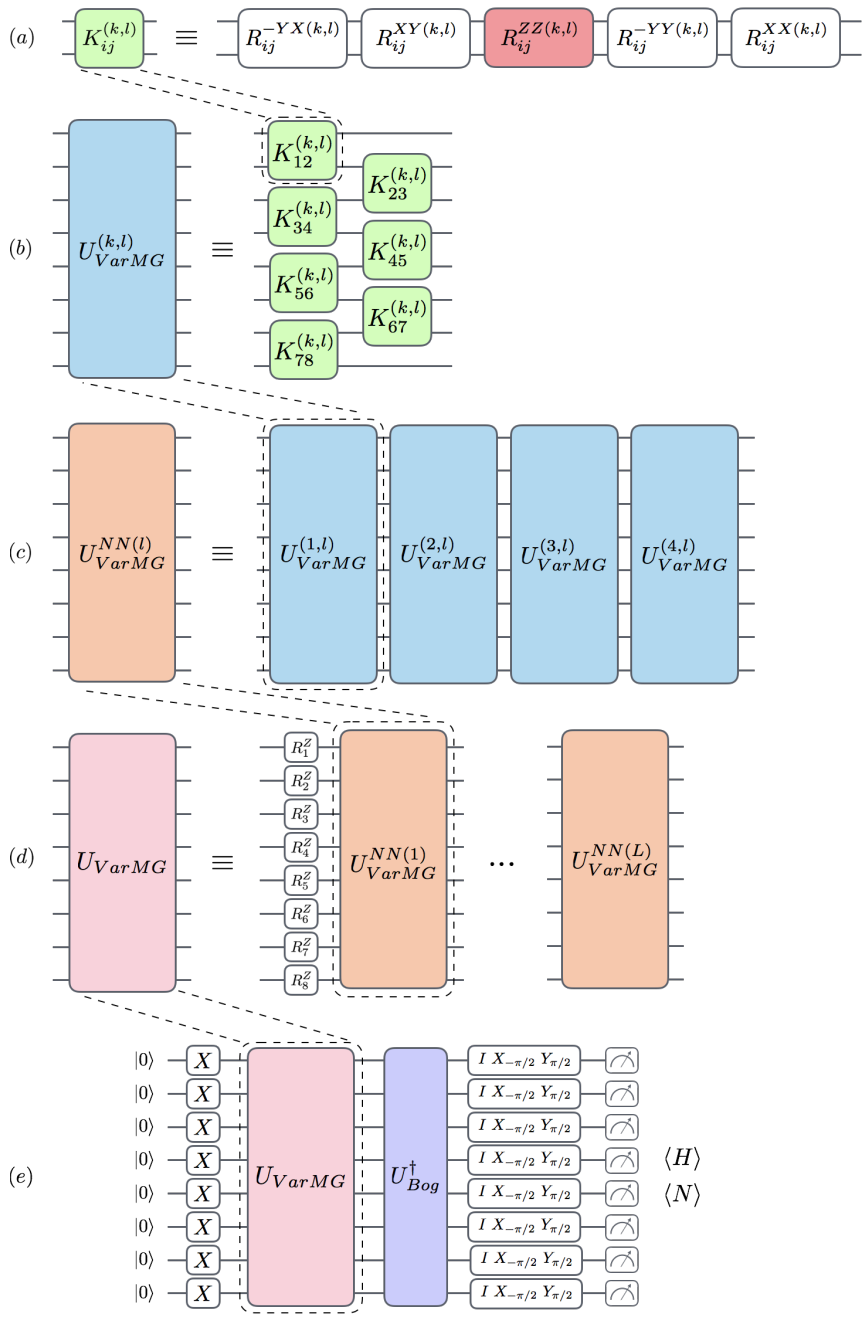


Figure 3.2: Gate decomposition of the L-cycle LDCA on a linear chain of 8 qubits. In (a), each $K_{ij}^{(k,l)}$ is a 2-local operation between qubits i and j composed of 5 rotations for a layer k. In (b), we build the unitary $U_{\mathrm{VarMG}}^{(k,l)}$ for each layer k by applying $K_{ij}^{(k,l)}$'s in parallel first on the even pairs and then on the odd pairs. In (c), a cycle $U_{\mathrm{VarMG}}^{\mathrm{NN}(l)}$ is composed by a sequence of $\lceil \frac{M}{2} \rceil$ layers. In (d), we show the L-cycle construction of U_{VarMG} with one round of variational phase rotations. The full LDCA protocol is shown in (e) with the initial preparation of the quasiparticle vacuum and the transformation to the original fermionic basis $U_{\mathrm{Bog}}^{\dagger}$.

The variational state therefore has the form

$$|\Psi(\Theta)\rangle = U_{\mathrm{Bog}}^{\dagger} U_{\mathrm{VarMG}}(\Theta) \prod_{i=1}^{M} X_i |0\rangle^{\otimes M}, \tag{3.46}$$

where it can be noticed that the $L = 0$ case is simply equivalent to producing the GHF state. There are 5 variational angles per $K_{ij}^{(k,l)}$ and $M-1$ of those terms per layer. Since each cycle has $\lceil \frac{M}{2} \rceil$ layers, a L-cycle circuit has $5L(M-1)\lceil \frac{M}{2} \rceil + M$ variational angles, the extra term arising from the round of phase rotations. Since gates can be operated in parallel in a linear chain of qubits, the circuit depth is $(10L+8)\lceil \frac{M}{2} \rceil + 4$ when we account for $U_{\mathrm{Bog}}^{\dagger}$ and the initial round of single-qubit X gates (this includes the final single-qubit rotations, $R_y(\frac{\pi}{2})$ or $R_x(-\frac{\pi}{2})$ gates (or equivalent), to measure the terms of the Hamiltonian in the form of Pauli strings). Therefore, this VQE scheme is hardware efficient in the sense that the circuit depth is linear in the number of qubits. The accuracy can also be systematically improved by increasing the number of cycles until either convergence is reached or errors dominate the precision of the result.

In the following section, we outline an implementation to compute the analytical gradient of the LDCA using quantum resources, which could be useful during the optimization procedure in VQE by guiding the search for the ground state and its energy.

3.2.5 Gradient Evaluation for LDCA

When optimizing the ansatz parameters to minimize the total energy, there may be a need to implement gradients depending on the selected optimization procedure. While direct search algorithms are generally more robust to noise than gradient-based approaches, they may require larger numbers of function evaluations [183]. On the other hand, numerical implementations of gradients rely heavily on the step size for accuracy. However, step sizes that are too small may lead to numerical instability and higher sampling cost. In addition, implementation of step sizes

corresponding to desired accuracy are limited by experimental errors.

An alternative approach that exhibits high accuracy while maintaining reasonable computational cost may be to evaluate the gradient directly on the quantum computer given that the analytical form of the gradient is available. Here we employ a scheme similar to one outlined in[279] tailored to implement the analytical gradient of the LDCA unitary using an extra qubit and controlled two-qubit rotations. Recall the unitary for the complete variational ansatz shown in (3.44), which we called $U_{VarMG}(\Theta)$ parametrized by angles Θ. For this derivation, we will ignore the products of Z-rotations in the definition but computing the gradient with respect to these angles should be more straightforward. These initial Z-rotations are not as "nested" within the LDCA framework, so the gradient corresponding to one of such angles, say θ_j, simply involves inserting a controlled-Z gate following the unitary $\exp(-i\theta_j Z)$, to the circuit (where we use an ancilla qubit as the control qubit). Thus, we will instead focus on finding the gradients of the term $\prod_{l=1}^{L} U_{VarMG}^{NN}(\Theta^{(l)})$, which we will call $U_{VarMG}^{'}(\Theta)$.

Consider the state $\Psi(\Theta)$, prepared by applying $U_{VarMG}(\Theta)$ to $|\Phi_0\rangle$, where $|\Phi_0\rangle$ corresponds to a reference state that does not depend on Θ. Here we wish to compute the derivative of the expectation value of the energy $E(\Theta) = \langle\Psi(\Theta)|H|\Psi(\Theta)\rangle$ with respect to each parameter in Θ. We will use the label $\theta_{j,n}^{(k,l)}$ for each parameter where j refers to the index of the qubit in the register, l to the circuit cycle, k to the circuit layer, and n to the appropriate Pauli string (in this case, $n \in \{-YX, XY, ZZ, -YY, XX\}$). Considering a Hamiltonian H that is independent of Θ, the derivative with respect to $\theta_{j,n}^{(k,l)}$ is given by

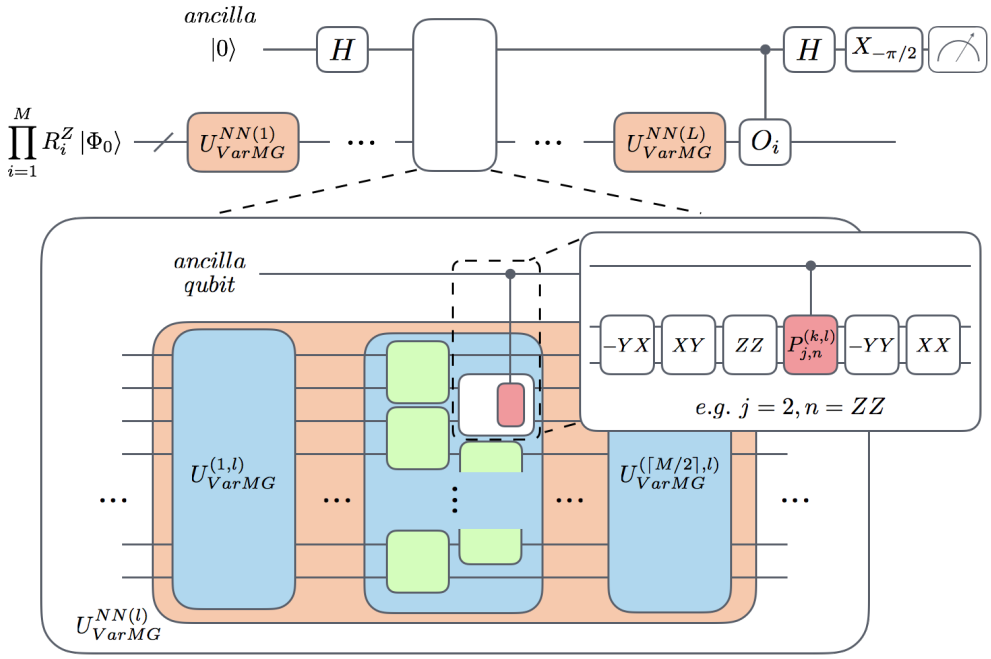


Figure 3.3: Circuit using an ancilla qubit to measure the imaginary component of $\langle\Phi_0|V_{j,n}^{(k,l)\dagger} O_i U'_{VarMG}|\Phi_0\rangle$ required to compute $\frac{\partial E(\Theta)}{\partial \theta_{j,n}^{(k,l)}}$. This figure illustrates an instance of the circuit where $j = 2$ and $n = ZZ$.

$$\frac{\partial E(\Theta)}{\partial \theta_{j,n}^{(k,l)}} = \langle\Phi_0|U^\dagger\, H\, \frac{\partial U}{\partial \theta_{j,n}^{(k,l)}}|\Phi_0\rangle + \langle\Phi_0|\frac{\partial U^\dagger}{\partial \theta_{j,n}^{(k)}}\, H\, U|\Phi_0\rangle \tag{3.47a}$$

$$= i\Big(\langle\Phi_0|U^\dagger\, H\, V_{j,n}^{(k,l)}|\Phi_0\rangle - \langle\Phi_0|V_{j,n}^{(k,l)\dagger}\, H\, U|\Phi_0\rangle\Big) \tag{3.47b}$$

$$= 2\,\mathrm{Im}\Big(\langle\Phi_0|V_{j,n}^{(k,l)\dagger}\, H\, U|\Phi_0\rangle\Big) \tag{3.47c}$$

where the operator $V_{j,n}^{(k,l)}(\Theta)$ is nearly identical to the unitary U'_{VarMG} except with a string of Pauli matrices $P_{j,n}^{k,l}$ inserted after the rotation term $R_{j,j+1}^{n(k,l)} = \exp(i\theta_{j,n}^{k,l} P_{j,n}^{k,l})$ included in the nearest-neighbor matchgate term $K_{j,j+1}^{(k,l)}$ and so on.

To compute the expectation value of the energy, we can employ the Hamiltonian averaging

procedure[222,226]. This involves measuring the expectation value of every term in the Hamiltonian and summing over them as shown in (3.48). Note that each term, which we call O_i, is a product of Pauli matrices obtained by performing the Jordan-Wigner or Bravyi-Kitaev transformation on the corresponding term in the second quantized Hamiltonian from (3.1).

$$E = \sum_i h_i \langle O_i \rangle. \tag{3.48}$$

Substituting (3.48) into (3.47c), the gradient can be expressed as:

$$\frac{\partial E(\Theta)}{\partial \theta_{j,n}^{(k,l)}} = 2\sum_i h_i \, \mathrm{Im}\Big(\langle \Phi_0 | V_{j,n}^{(k,l)\dagger}(\Theta) O_i U(\Theta) | \Phi_0 \rangle \Big) \tag{3.49}$$

Each of the terms in the sum above can be computed using the circuit shown in Figure 3.3. For a practical physical implementation of the analytical gradient, a circuit layout similar to one highlighted in[80] could be used, in which the control qubit of the gradient circuit is connected to all qubits in the register.

In the following section, we numerically benchmark the BUCC ansatz and LDCA on small instances of the Fermi-Hubbard model and the automerization reaction of cyclobutadiene, where we find that LDCA is able to prepare the exact ground state of those systems.

3.3 Numerical examples

In this section, we numerically test the performance of the previously described algorithms on instances of strongly correlated systems in condensed matter and quantum chemistry. Specifically, in subsection 3.3.1, we analyze the behavior of the ansatz on the Fermi-Hubbard model at half-filling at different interaction strengths. In subsection 3.3.2, we study the automerization reaction of cyclobutadiene modeled using the Pariser-Parr-Pople (PPP) Hamiltonian[251,252,261]. In

both cases, the Hamiltonians are mapped to 8-qubit registers and we compare the energy and wavefunction accuracies for approximating the exact ground state for the following methods ansatzes: GHF, BUCCSD, and LDCA with 1 and 2 cycles.

In these cases, the state initialization has 8 single qubit X gates operated in parallel and the inverse Bogoliubov transformation has one layer of single qubit phase rotations and 112 nearest-neighbor matchgates. The state initialization and $U_{\text{Bog}}^{\dagger}$ circuit add up to a circuit depth of 34. The LDCA method adds a layer of variational phase rotations and 140 nearest-neighbor gates per cycle. Therefore 1-cycle LDCA adds 41 to the circuit depth (for a total of 75 with 148 variational parameters) and 2-cycle LDCA adds 81 to the circuit depth (for a total of 115 with 288 variational parameters).

For the numerical examples presented here, we find that 2-cycle LDCA is able to exactly recover the ground state of the simulated systems while 1-cycle LDCA performs better than the GHF solution but is not as accurate as BUCCSD. An important caveat is that the 2-cycle LDCA has more variational parameters (288) than the dimensions of the Hilbert space ($2^8 = 256$) but we still consider the result relevant for experimental implementations since the depth of the circuit is much shorter than what could be achieved with BUCC up to fourth order, which is required to recover the exact ground state of systems studied.

3.3.1 Fermi-Hubbard model

The Fermi-Hubbard model[152] is a prototypical example of correlated electrons. It is described by a tight-binding lattice of electrons interacting through a local Coulomb force. The Hamiltonian is given by

$$\begin{aligned} H^{\text{FH}} &= -t\sum_{\langle p,q\rangle}\sum_{\sigma=\uparrow,\downarrow}\left(a_{p\sigma}^{\dagger}a_{q\sigma}+a_{q\sigma}^{\dagger}a_{p\sigma}\right) \\ &\quad -\mu\sum_{p}\sum_{\sigma=\uparrow,\downarrow}\left(n_{p\sigma}-\tfrac{1}{2}\right) \\ &\quad +U\sum_{p}\left(n_{p\uparrow}-\tfrac{1}{2}\right)\left(n_{p\downarrow}-\tfrac{1}{2}\right), \end{aligned} \tag{3.50}$$

where t is the kinetic energy between nearest-neighbor sites $\langle p,q\rangle$, U is the static Coulomb interaction and μ is the chemical potential. The number operator is $n_{p\sigma} = a^\dagger_{p\sigma} a_{q\sigma}$. While the one-dimensional Fermi-Hubbard model can be solved exactly with the Bethe ansatz[102,204], the two-dimensional version can only be solved exactly for very specific values of the parameters and a general solution remains elusive. The phase diagram of the 2D model is known to be very rich and there are strong arguments that a better understanding of the model could yield the key to explain the physics of high-temperature cuprate superconductors[7,8,196].

Hybrid quantum-classical methods to systematically approximate the phase diagram of the Fermi-Hubbard model in the thermodynamical limit are known[79,80] but they require preparing the ground state of a large cluster of the model with an accuracy that cannot be reached by previously proposed methods[355]. Here, we investigate the performance of the ansatz detailed in section 3.2 on an example of a 2×2 cluster of the Fermi-Hubbard model at half-filling ($\mu = 0$) that can be implemented on a 8-qubit quantum processor. As shown in figure 3.4, the GHF method performs well for small values of the interaction strength $\frac{U}{t}$ and exactly describes the tight-binding case where the Hamiltonian is quadratic. The BUCCSD ansatz offers a significant improvement over the GHF solution but fails to reach the exact ground state at strong interaction strengths. While 1-cycle LDCA offers an intermediate solution between GHF and BUCCSD, the 2-cycle LDCA solution performs surprisingly well as it is able to reach the exact ground state up to numerical accuracy for all values of the interaction strength. In all cases the preparation fidelity $|\langle \Psi(\Theta) | \Psi_0 \rangle|^2$ is directly correlated with the energy difference δE between the prepared state and the exact ground state $|\Psi_0\rangle$. We also show that all methods are able to handle Hamiltonians with pairing terms by introducing an artificial $\Delta \sum_i \left(a^\dagger_{i\uparrow} a^\dagger_{i\downarrow} + a_{i\downarrow} a_{i\uparrow} \right)$. The accuracy of all methods improves with increasing $\frac{\Delta}{t}$ as the ground state gets closer to a fermionic Gaussian state.

We also tested a simpler one dimensional cluster of the Fermi-Hubbard model with 2 sites and

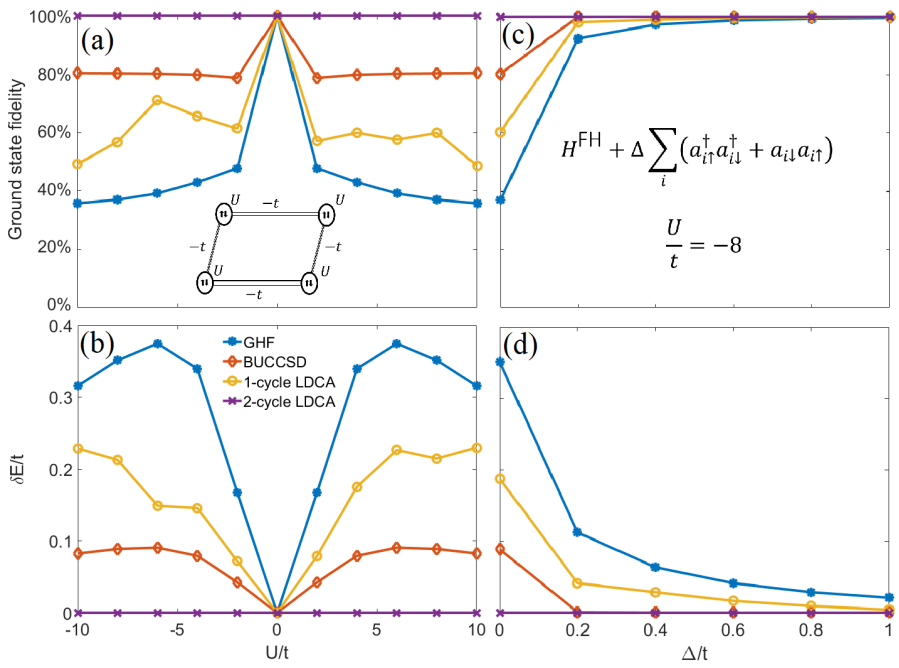


Figure 3.4: In (a), we show the fidelity of the ground state preparation of a 2×2 cluster of the Fermi-Hubbard model as a function of the interaction parameter U. The energy difference with the exact ground state with respect to the various methods is shown in (b). The energies are normalized by the hopping term t. In (c) and (d), we show respectively the fidelity and the energy difference in the case of an attractive cluster $\frac{U}{t} = -8$ with an additional s-wave pairing term Δ.

found that it was possible to reach the exact ground state with both BUCCSD and the 1-cycle LDCA method for all values of the parameter U. This is expected for the BUCCSD method as this is equivalent to a full configuration interaction parametrization in this specific case. We do not have sufficient information to determine the number of cycles L required by LDCA to reach the ground state as a function of the cluster size since it would require much more intense numerics. However, the fact that a 2×1 cluster requires only 1 cycle and that the 2×2 case reaches the ground state in 2 cycles leave open the possibility that the scaling is not an exponential function of the cluster size.

3.3.2 Cyclobutadiene

As an example of a quantum chemistry application, we studied the accuracy of the proposed methods in the description of cyclobutadiene automerization. The study of this reaction has been

particularly challenging for theoretical chemists due to the strongly correlated character of the open-shell D_{4h} transition state in contrast with the weakly correlated character of the closed-shell D_{2h} ground state ($^1A_{1g}$)[328]. An accurate theoretical treatment of the transition state would allow to confirm several observations about the mechanism, such as the alleged change in the aromatic character of the molecule between its ground and transition states as well as the involvement of a tunneling carbon atom in the reaction[10–12,328]. In addition, it would serve as a confirmation of the energy barrier for the automerization, for which experimental reports vary between 1.6 and 12.0 kcal/mol[360].

Although the Hamiltonian for cyclobutadiene can be obtained from a Hartree-Fock or a Complete Active Space (CAS) standard quantum chemistry calculation, we opted to describe the reaction using a Pariser-Parr-Pople (PPP) model Hamiltonian[251,252,261]. The PPP model captures the main physics of π-electron systems such as cyclobutadiene and also establishes a direct connection to the Fermi-Hubbard Hamiltonian studied in the previous section. Using this model, the Hamiltonian of cyclobutadiene can be written as

$$\begin{aligned} H^{\mathrm{PPP}} &= \textstyle\sum_{i<j} t_{ij} E_{ij} + \sum_i U_i n_{i\alpha} n_{i\beta} + V_c \\ &\quad + \tfrac{1}{2} \textstyle\sum_{ij} \gamma_{ij} (n_{i\alpha} + n_{i\beta} - 1)(n_{j\alpha} + n_{j\beta} - 1), \end{aligned} \tag{3.51}$$

where $E_{ij} = \sum_{\sigma=\alpha,\beta} a^\dagger_{i\sigma} a_{j\sigma} + a^\dagger_{j\sigma} a_{i\sigma}$, $n_{i\sigma} = a^\dagger_{i\sigma} a_{i\sigma}$, and the variables γ_{ij} are parameterized by the Mataga-Nishimoto formula[220]

$$\gamma_{ij}(r_{ij}) = \frac{1}{1/U + r_{ij}}. \tag{3.52}$$

The t_{ij}, U, and V_c parameters were obtained from[290,291] as a function of the dimensionless reaction coordinate, λ, and the geometries of the ground as well as transition states were optimized at this level of theory.

Figure 3.5 compares the accuracies of different ansatzes for the cyclobutadiene automerization

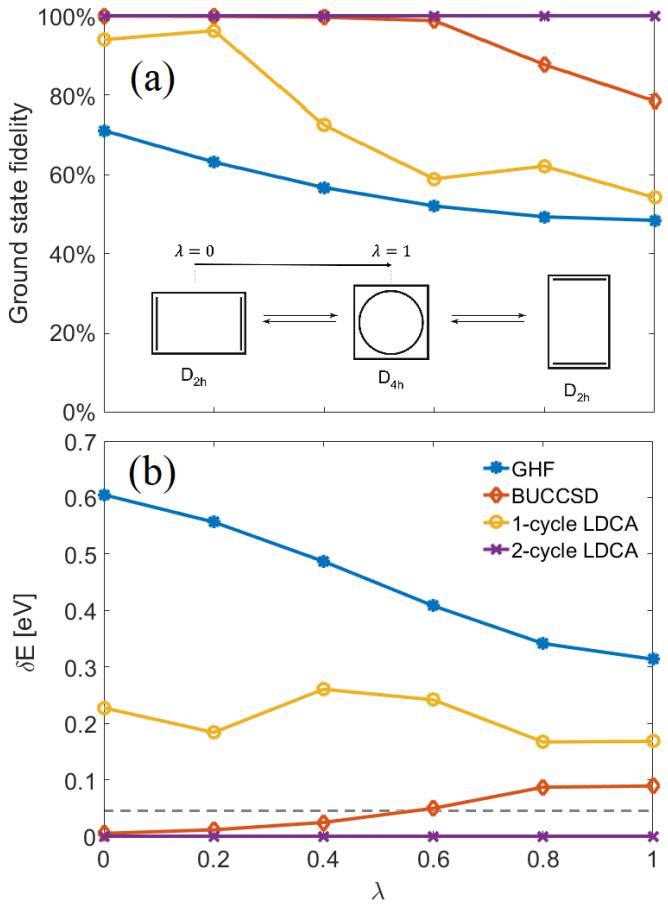


Figure 3.5: In (a), we show the fidelity of the ground state preparation along the automerization reaction path of cyclobutadiene. Subfigure (b) shows the difference with the exact ground state for the various ansatzes. Chemical accuracy is approximately 0.043 eV (dashed line).

reaction. We observe that GHF ansatz is considerably improved by BUCCSD close to the D_{2h} ground state but the improvement is less prominent as we approach the strongly correlated D_{4h} transition state. As in the 2×2 Fermi-Hubbard case, the 1-cycle LDCA method yields accuracies between those of GHF and BUCCSD while the 2-cycle LDCA method produces the numerically exact ground state for all values of λ. This surprising result suggests that LCDA is potentially useful for treating cases of strong correlation in quantum chemistry.

3.4 Discussion

The results presented in the previous section suggest that the LDCA could outperform other ansatzes employed for VQE calculations, such as BUCC, both in accuracy and efficiency. Being a method inspired by BUCC, the LDCA scheme inherits some properties of this ansatz. For instance, in the limit of 1-cycle LDCA with all $\theta_{ij}^{ZZ(k,1)}$ set to zero, we recover the BUCC ansatz with single excitations. We point out that this choice of parameters cannot improve the GHF solution since it only amounts to a basis rotation of the fermionic mode for which the Bogoliubov transformation has already been optimized. Since the mapping (3.35) between the Bogoliubov transformation and the matchgate circuit relies on the Jordan-Wigner transformation which associates Pauli strings of length $O(M)$ to fermionic operators, it may be possible to further reduce the length of the measured Pauli strings by working out a similar mapping in the Bravyi-Kitaev basis[48] where operators are represented by strings of length $O(\log M)$. For completeness, we also numerically benchmarked the traditional UCCSD scheme[226,255,279] and found that it provides the same results as BUCCSD. This is expected in the case of Hamiltonians with no explicit pairing terms. However, such terms may appear in variational self-energy functional theory[79,262,263,301,302] where fictitious pairing terms are added to a cluster Hamiltonian to recover the magnetic and superconducting phase diagram in the thermodynamic limit.

Regarding the number of variational parameters, LDCA scales as $O\left(LM^2\right)$ compared to $O\left(M^4\right)$ for UCCSD and BUCCSD with Gaussian basis set. There may exist constraints on the variational parameters of LDCA that reduce their total number. To explore whether it was possible to only measure $\langle H \rangle$ in the variational procedure, we tried the ansatz with only number conserving terms (such that all $\theta_{ij}^{XY(k,l)} = \theta_{ij}^{-YX(k,l)} = 0$) on the Fermi-Hubbard model but found a reduced overlap with the exact ground state. This implies that a reconfiguration of the pairing amplitudes with respect to the GHF reference state is an important condition to reach an accurate ground state.

Our estimates of the circuit depth assume a quantum architecture consisting of a linear chain of qubits, which allows to maximize the parallel application of gates through the algorithm. We leave open the question of whether it is possible to achieve further improvements by using an architecture with increased connectivity. We also assumed that nearest-neighbor two-qubit gates could be implemented directly (as proposed for a linear chain of polar molecules[144]). Although this is not the case on current ion trap and superconducting circuit technologies, the required gates can be implemented as long as tunable nearest-neighbor entangling gates are available. In this case, only additional single-qubit basis rotation suffices, adding only a a small overhead in circuit depth[243].

Due to its better accuracy and reduced scaling in depth and number of parameters compared to previous ansatzes, we believe that the LDCA approach is a feasible alternative for studying strongly correlated systems in near-term quantum devices. In this case, we propose some strategies to ensure a better performance of the ansatz on real quantum processors with control inaccuracies. For instance, we could calibrate the angles $\theta_{ij}^{\mu\nu}$ of the gate sequence of $U_{\mathrm{Bog}}^{\dagger}$ by minimizing the difference between the values of $\langle H \rangle$ and $\langle N \rangle$ measured on the quantum computer and the values obtained numerically for the GHF reference state. Similarly, it should be possible to experimentally estimate the errors on the energy and the number of particles for a given L-cycle LDCA by comparing the values of $\langle H \rangle$ and $\langle N \rangle$ obtained with all $\theta_{ij}^{ZZ(k,1)}$ set to zero with the exact classical results computed as described in section 3.2.3. Instead of setting $\theta_{ij}^{ZZ(k,1)}$ to zero, one might also replace the ZZ rotations with equivalent time delays.

Finally, we point out that our formalism should be general enough to implement the simulation of nucleons[310]. However, we abstained from venturing in the numerical simulation of such systems as it is beyond our fields of expertise. Similarly, our method could be employed to study the ground state of gauge theories in the quantum link model[56,81,375].

3.5 Conclusion

In this work, we generalized the Bogoliubov coupled cluster ansatz to a unitary framework such that it can be implemented as a VQE scheme on a quantum computer. We showed how the required GHF reference state can be computed from the theory of fermionic Gaussian states. Those states include Slater determinants used in quantum chemistry as well as mean field superconducting BCS states. We described a procedure to prepare fermionic Gaussian states on a quantum computer using a circuit of nearest-neighbor matchgates with linear depth on the size of the system. By augmenting the set of available gates with nearest-neighbor $\sigma_z \otimes \sigma_z$ rotations, we constructed a low-depth circuit ansatz (LDCA) that can systematically improve the preparation of approximate ground states for fermionic Hamiltonians. Each added cycle increases linearly the depth of the quantum circuit, which makes it practical for implementations in near-term quantum devices.

We used a cluster of the Fermi-Hubbard model and the automerization of Cyclobutadiene as examples to assess the accuracy of the BUCC and LDCA ansatzes. Our results showed that LDCA has the potential to accurately described the exact ground state of strongly correlated fermionic systems on a quantum processor. In addition, our proposed BUCC and LDCA approaches can be used to approximate the ground states of Hamiltonians with pairing fields. This feature, not present in previous ansatzes such as UCC, extends the range of applicability of VQE to problems in condensed matter and nuclear physics. Since the number of particles is not conserved in BUCC and LDCA, we must impose constraints on the number of particles to carry out the optimization in the classical computer. Future work will be devoted to benchmarking the accuracy of the LDCA method for a larger variety of molecular systems and determining the scaling in the number of cycles required to describe the ground states of general systems.

4

Towards computing molecular energies on near-term quantum devices

The results presented in this chapter originally appeared in the following papers[142,248]:

"Scalable Quantum Simulation of Molecular Energies". Peter O'Malley, Ryan Babbush, Ian Kivlichan, Jonathan Romero, Jarrod R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding et al. *Phys. Rev. X.*, **6**, 031007 (2016). Copyright (2018) by American Physical Society.

"Quantum Chemistry Calculations on a Trapped-Ion Quantum Simulator". Cornelius Hempel, Christine Maier, Jonathan Romero, Jarrod R. McClean, Thomas Monz, Heng Shen, Petar Jurcevic, Ben P. Lanyon, Peter Love, Ryan Babbush, Alán Aspuru-Guzik, Rainer Blatt and Christian F. Roos. *Phys. Rev. X.*, **8**, 031022 (2018). Copyright (2018) by American Physical Society.

4.1 Introduction

Quantum simulation is one of the most compelling applications for quantum computers[117]. A quantum simulator aims to utilize controlled quantum evolutions to simulate the behavior of a quantum system of interest. One approach to building such a machine is to engineer a quantum system to closely match the interactions in the Hamiltonian of the system of interest, an approach known as analog simulation. An alternative approach, known as digital simulation, employs a set of quantum operations or gates that can implement the time evolution of arbitrary Hamiltonians. Consequently, a quantum simulator manipulated in a programmable fashion[210] can implement error correction[203,332], becoming equivalent to a quantum computer. The most basic advantage of a

quantum simulator compared to a classical computer is its ability to *efficiently* represent a quantum system. In contrast, storing and manipulating arbitrary quantum states on a classical computer requires exponential resources. While certain quantum systems admit efficient classical approximations, many problems of interest in Chemistry and material sciences, for example, still require daunting resources on a classical computer[117], establishing quantum simulators as valuable tools for researchers in many fields.

Among the disciplines that can benefit from a quantum simulator, Chemistry has perhaps the highest impact due to its wide range of industrial applications that include the design of new drugs, materials, catalysts, among other types of molecules of high industrial value[61,148]. This prospect has motivated the development of a variety of algorithms for simulating chemical phenomena with quantum computers[62,368]. Most of these approaches belong to two categories: algorithms for error-corrected quantum computers, based on Hamiltonian simulation techniques, and variational algorithms. For a given Hamiltonian H, Hamiltonian simulation is an efficient and accurate implementation on a quantum computer of a quantum operation which has eigenvalues that are a known function of the eigenvalues of H, e.g. e^{-iHt}. In practice, this typically results in quantum circuits with a significant number of gates, which demands access to an error-corrected quantum computer.

Alternatively, variational algorithms exploit the variational principle to compute approximations to the ground state[226] or the dynamics[202] of a quantum system using a quantum circuit as a variational ansatz. Consequently, these approaches can be implemented using relatively short depth circuits as ansatzes. Besides, they have shown robustness against certain types of errors[226]. These features enable variational algorithms to be implemented even without quantum error correction, positioning them to as the *de facto* practical algorithm for near-term quantum devices.

Currently, quantum devices based on trapped ions[134,235] and superconducting qubits[356]

incorporate several qubits and can perform quantum gates with fidelities nearing 99%. Such machines constitute the foundation of intermediate-scale quantum (NISQ) computers[269], which are programmable arrays of tens to hundreds of qubits, capable of executing circuits with depths in the order of thousands of elementary two-qubit operations[269]. Such specifications provide a perfect test bed for variational algorithms applied to Chemistry simulation. Naturally, to make progress in this direction, it is indispensable to test these algorithms on existing quantum hardware, such that they can be improved and adapted to the ever-evolving quantum machines.

In this Chapter, we summarize the results of two proof of principle implementations of the variational quantum eigensolver (VQE) algorithm on prototype quantum computing devices[142,248]. As described in Chapter 1, the VQE algorithm computes the ground state of a quantum system using the time-independent variational principle[226]. More specifically, VQE postulates a quantum circuit as variational ansatz and computes an approximation to the ground state of the Hamiltonian of interest by minimizing the expectation value of this Hamiltonian with respect to the circuit parameters[255]. In the first experiment, we implement VQE to simulate the ground state of molecular hydrogen (H_2) at different internuclear separations on a reconfigurable array of superconducting qubits. In the second experiment, we implement VQE to simulate molecular hydrogen and lithium hydride (LiH) on a multi-qubit trapped ion device. These experiments constitute the first scalable simulations of chemistry using these types of quantum computing architectures, respectively. *

This chapter is organized as follows: in Section 4.2, we review the steps of the VQE algorithm and describe details of the quantum devices employed in the experiments. We also present in detail the classical pre-computations needed for the implementation of VQE for H_2 and LiH. In Section 4.3, we present the results and analysis of the proof of principle experiments for both devices. In

*In the context of quantum computing, scalable implementations refer to an implementation where the required classical and quantum resources increase in the same way that they would for an arbitrarily large problem.

Section 4.4, we analyze the significance of these results and offer some final remarks.

4.2 VQE IMPLEMENTATION ON NEAR-TERM QUANTUM DEVICES

Figure 4.1 summarizes the algorithmic steps for implementing VQE on a quantum computer, which we described in detail for a UCC ansatz in Section 2.2. We follow this procedure to perform VQE calculations of two small chemical systems, molecular Hydrogen (H_2) and Lithium hydride (LiH), using two different quantum processors. In our experimental demonstrations, we describe the chemical systems using a minimal basis set (STO-nG). In this case, each Hydrogen atom is assigned two Slater-type orbitals with angular momentum zero ($l = 0$), each of them approximated as a combination of n primitive Gaussian-type orbitals (GTOs). Similarly, each Lithium atom is described using 5 Slater-type orbitals, each of them approximated as a linear combination of GTOs. Although such small basis sets do not provide property predictions within chemical accuracy of experimental values, they serve to illustrate the implementation of the algorithm. A Hartree-Fock calculation executed on a classical computer provides the optimal molecular orbitals used for building the corresponding Hamiltonians in second quantization.

To quantify the accuracy of the results, we employ the absolute difference between the energy calculated with the VQE procedure and the energy corresponding to the exact diagonalization of the Hamiltonian. We also compare relative energy differences calculated along the PES. We performed experiments using two different quantum platforms: 1) a programmable array of superconducting qubits, and 2) a programmable array of trapped ions. We will designate these devices as SQD (superconducting qubit device) and TID (trapped ion device). Both devices fulfill the DiVicenzo's criteria[93], described in Section 1.1, and therefore correspond to quantum computers. However, they do not implement error correction.

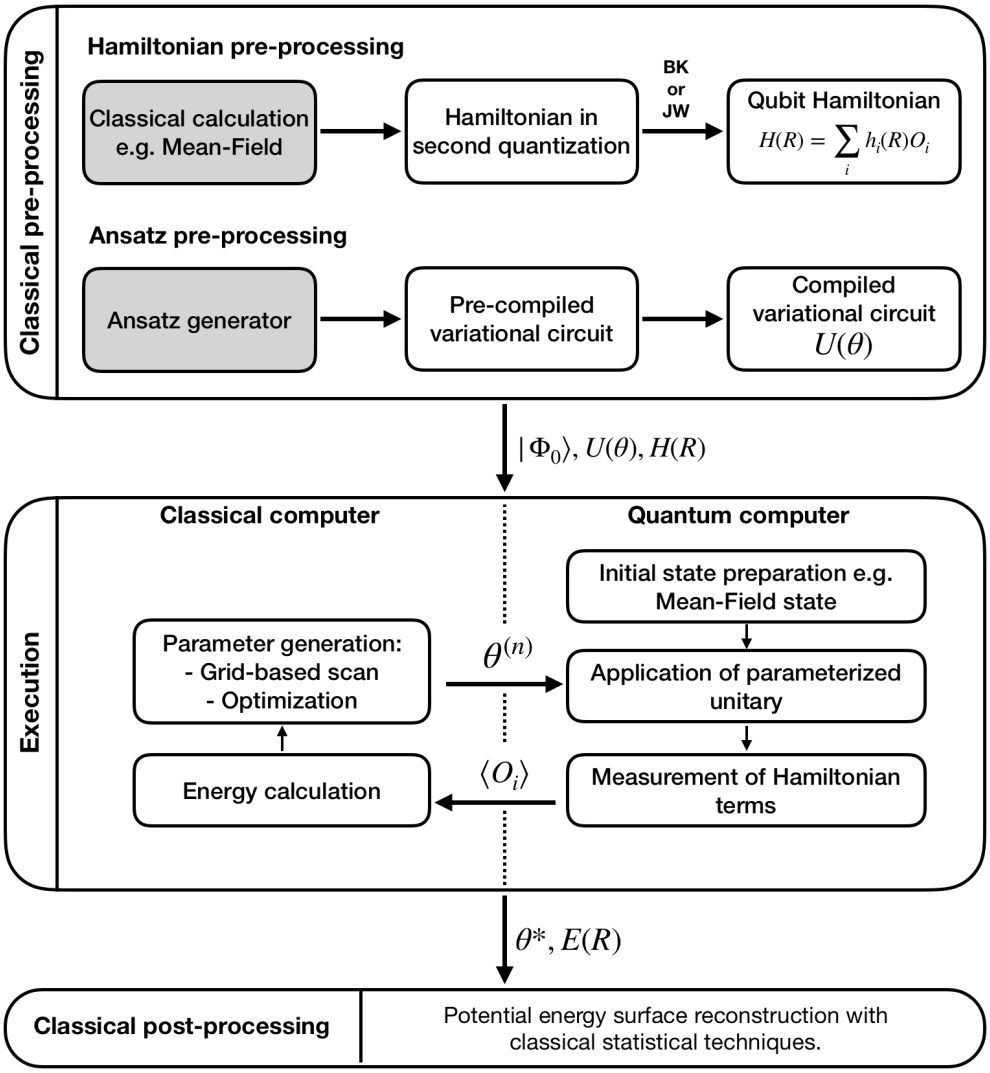


Figure 4.1: Steps of a VQE-based quantum chemistry calculation: 1) In the classical pre-processing step, the molecular Hamiltonian is generated and subsequently mapped into a qubit Hamiltonian using the Bravyi-Kitaev (BK) or Jordan Wigner (JW) transformations. This step also involves the generation of the variational circuit ($U(\theta)$) and the initial state ($|\Phi_0\rangle$) and its compilation to the native gate set and architecture of the quantum processor employed for the calculation. 2) The execution of the algorithm comprises a hybrid quantum-classical feedback loop, where the quantum computer is employed to execute the state preparation and measurement of the expectation value of the target Hamiltonian, while the classical computer computes the energy and propose a new set of variational parameters using a classical optimization routine. 3) The results obtained from the variational loop are post-processed to reconstruct the potential energy surface of the target chemical system.

4.2.0.1 DESCRIPTION OF THE QUANTUM DEVICES

In the superconducting qubit architecture, the energy levels of an artificial atom realized with an electric LC circuit encode the qubits. In superconducting conditions, the charge carriers are Cooper pairs (pairs of electrons) which condense into a macroscopic quantum state. The collective motion of this condensate is described by the flux threading the inductor, which plays the role of the center-of-mass position in a mass-spring mechanical oscillator, as described in[90]. An electrical component known as Josephson tunnel junction provides a non-dissipative and non-linear inductance that creates an anharmonic potential, generating an artificial atom with addressable states [†]. Excitation energies in these devices are typically in the microwave range. Varying the inductance, capacitance, and properties of the junction, it is possible to realize different potentials. Furthermore, a superconducting qubit can be coupled to photons in a microwave cavity (microwave resonator) to accomplish single-qubit gates and perform qubit readout. Several qubits can be coupled using intermediate coupling circuit components, such as a capacitor, or through additional microwave cavities, to realize two-qubit gates.

In our experiments, we employed a linear array of Xmon qubits[22]. The Xmon is a variant of a charge qubit where different energy levels correspond to a different integer number of Cooper pairs. As described in[248], the qubit array is kept in a dilution refrigerator with a base temperature of 20 mK. Each qubit consists of a superconducting quantum interference device (SQUID), which provides a tunable nonlinear inductance, and a large X-shaped capacitor; the qubits are capacitively coupled to their nearest neighbors in a linear chain pattern. Single-qubit quantum gates are implemented with microwave pulses and tuned using closed-loop optimization with randomized benchmarking[175]. Qubit state measurement is performed in a dispersive readout scheme with capacitively coupled resonators. Details on the device fabrication are provided in[176].

[†]As the potential becomes anharmonic, the energy difference between states is no longer the same as in a harmonic oscillator. This allows addressing a pair of states selected as computational states

The entangling operation in the SQD architecture is a controlled-phase (CZ_ϕ) gate, which has the form:

$$CZ_\phi = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{-i\frac{\phi}{2}} & 0 \\ 0 & 0 & 0 & e^{i\frac{\phi}{2}} \end{bmatrix}. \tag{4.1}$$

The CZ_ϕ gate is accomplished using an adiabatic qubit frequency tuning procedure described in[23]. The CZ_ϕ gate is implemented in a range of 0.25-5.0 rad; for ϕ outside this range, the total rotation is accomplished with two physical gates. Details of the implementation of the circuit for VQE and experimental conditions of the execution can be found in Ref.[248].

In the case of trapped ion architectures, two addressable electronic states or Zeeman states of isolated atomic ions encode a qubit. As described in[142], the trapped ion system employed in our experiments consists of a linear Paul trap in which a variable number of $^4Ca^+$ ions are confined using an electrical field. Each atom encodes a qubit in a pair of Zeeman states chosen from the $4S_{1/2}$ electronic ground and $3D_{5/2}$ metastable states, corresponding to states $|1\rangle$ and $|0\rangle$, respectively. The qubits are manipulated via a set of global, tightly focused, addressed laser beams. Single-qubit gate operations are implemented via a three-pulse sequence that combines two global qubit rotations with an intermediate addressed laser pulse manipulating only the targeted qubit[289]. Multi-qubit entangling operations are realized through laser-driven interactions that are mediated by the collective motional modes of the ions within their common trapping potential.

The setup described above provides the elements necessary to implement the following universal

set of quantum gates[192]:

$$R_r^j(\theta) = exp\left(-i\frac{\theta}{2}\sigma_R^j\right) \quad r \in \{x,y,z\} \quad \text{(Single qubit rotations);} \tag{4.2}$$

$$U_2^S(\theta) = exp\left(-i\frac{\theta}{2}\sum_{j\in S}\sigma_z^j\right) \quad \text{(Collective single qubit gate);} \tag{4.3}$$

$$U_3^S(\theta,\phi) = exp\left(-i\frac{\theta}{2}\sum_{j\in S}\sigma_\phi^j\right) \quad \text{(Global carrier rotation);} \tag{4.4}$$

$$MS^S(\theta,\phi) = exp\left(-i\frac{\theta}{2}\sum_{\{j<k\}\in S}\sigma_\phi^j\sigma_\phi^k\right) \quad \text{(MS entangling gate);} \tag{4.5}$$

where σ_k^j denotes the k-th Pauli matrix acting on the j-th qubit and $\sigma_\phi^j = cos(\phi)\sigma_x^j + sin(\phi)\sigma_y^j$. S represents the subset of ions on which the gate acts. Mølmer-Sørensen (MS) gates on subsets of ions can be achieved either by applying hiding pulses, that transfer ions into decoupled electronic levels, or by employing refocusing techniques[238]. In our implementation, $\phi = 0$, such that our MS gates correspond to evolution under a Hamiltonian of pairwise XX interactions. The MS gate with $\theta = \frac{\pi}{2}$ corresponds to a fully-entangling gate as it prepares a maximally entangled state in the subset S of qubits in the initial state $|1\rangle^{\otimes n}$.

4.2.0.2 Error sources in the VQE implementation

The quantum devices used in our experiments do not implement quantum error correction. Instead, operations are executed directly on physical qubits and are subject to noise processes arising from the interaction with the environment and quantum control imperfections, e.g., laser fluctuations coupling to the motion of the ions in TID or parasitic interactions among the qubits in SQD. These noise processes are usually characterized in terms of the energy relaxation (T_1) and dephasing (T_2) times of the qubits and fidelities of the quantum gates.

Energy relaxation quantifies the time it takes for a qubit to decay from its excited state $|1\rangle$ to the ground state $|0\rangle$ (a bit-flip error) while dephasing times correspond to the time it takes for a

quantum superposition state $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ to lose its phase relationship between $|0\rangle$ and $|1\rangle$ (a phase-flip error). Shorter times indicate reduced accuracy of the quantum operations. The gate fidelity measures the overlap between the quantum channel employed to implement a quantum gate in a specific architecture and its ideal unitary channel. Correspondingly, fidelity values range from 0 to 1 (0 to 100%), with 1 (100%) corresponding to perfect overlap.

In addition to environmental noise, the information extracted from the quantum computer is also subject to fluctuations in the statistics of the measurements collected from the device. This error, known as quantum projection noise (QPN), limits the precision ε to which we can estimate the expectation values required in the VQE algorithm, as described in Section 2.3.3. The QPN scales as $O(1/\sqrt{M})$, where M is the number of measurements taken per data point.

4.2.1 H_2 IN A MINIMAL BASIS

The molecular Hamiltonian for H_2 in the STO-nG minimal basis can be mapped from its Fermionic form to qubits yielding:

$$\begin{aligned} H^{\mathrm{BK}} =& f_0\mathcal{I} + f_1\sigma_0^z + f_2\sigma_1^z + f_3\sigma_2^z + f_4\sigma_1^z\sigma_0^z + f_5\sigma_2^z\sigma_0^z + f_6\sigma_3^z\sigma_1^z + f_7\sigma_2^x\sigma_1^z\sigma_0^x + f_8\sigma_2^y\sigma_1^z\sigma_0^y + f_9\sigma_2^z\sigma_1^z\sigma_0^z \\ &+ f_{10}\sigma_3^z\sigma_2^z\sigma_0^z + f_{11}\sigma_3^z\sigma_2^z\sigma_1^z + f_{12}\sigma_3^z\sigma_2^x\sigma_1^z\sigma_0^x + f_{13}\sigma_3^z\sigma_2^y\sigma_1^z\sigma_0^y + f_{14}\sigma_3^z\sigma_2^z\sigma_1^z\sigma_0^z, \end{aligned} \tag{4.6}$$

where the coefficients f_i depend on the internuclear separation (R) between the hydrogen atoms and are derived from the molecular integrals obtained from the Hartree-Fock calculation. To perform such calculations we employed the OpenFermion library available in Python[225].

The reference state for the calculation corresponds to the Hartree-Fock solution in the molecular orbital basis, which in the Jordan Wigner mapping corresponds to a computational state $|\varphi_{\mathrm{HF}}\rangle = |0001\rangle$. We observe that the terms in the Hamiltonian only act with the identity and σ^z operations on qubits 1 and 3. This fact allows us to rewrite our reference state as

$|\varphi_{\mathrm{HF}}\rangle = |0\rangle_1|0\rangle_3 \otimes |0\rangle_2|1\rangle_0$. As qubits 1 and 3 will not experience population changes under this Hamiltonian, we can reduce Eq. (4.6) to an effective Hamiltonian acting on two qubits, with reference state $|\varphi_{HF}\rangle = |01\rangle$:

$$H^{\mathrm{BK}} = c_0\mathcal{I} + c_1\sigma_0^z + c_2\sigma_1^z + c_3\sigma_0^z\sigma_1^z + c_4\sigma_0^x\sigma_1^x + c_5\sigma_0^y\sigma_1^y, \tag{4.7}$$

where we have relabeled qubits 0 and 2 as 0 and 1. The coefficients c_i are given by:

$$\begin{aligned} c_0 &= f_0 + f_2 + f_6 & c_3 &= f_5 + f_9 + f_{10} + f_{14} \\ c_1 &= f_1 + f_4 & c_4 &= f_7 + f_{12} \\ c_2 &= f_3 + f_{11} & c_5 &= f_8 + f_{13} \end{aligned} \tag{4.8}$$

This reduction of the problem for the hydrogen molecule was first noted in[248], developed into a general method in Ref.[47] and used in superconducting implementations of several problems in Ref.[168], as accounted in[142]. All the required expectation values of the BK H_2 Hamiltonian can be measured with three different sets of projective measurements.

In contrast, the molecular Hamiltonian under the Jordan-Wigner transformation gets mapped to:

$$\begin{aligned} H^{\mathrm{JW}} =& c_0\mathcal{I} + c_1(\sigma_0^z + \sigma_1^z) + c_2(\sigma_2^z + \sigma_3^z) + c_3\sigma_3^z\sigma_2^z + c_4\sigma_1^z\sigma_0^z + c_5(\sigma_2^z\sigma_0^z + \sigma_3^z\sigma_1^z) + c_6(\sigma_2^z\sigma_1^z + \sigma_3^z\sigma_0^z) \\ &+ c_7(\sigma_3^x\sigma_2^y\sigma_1^y\sigma_0^z + \sigma_3^y\sigma_2^x\sigma_1^x\sigma_0^y) - c_7(\sigma_3^x\sigma_2^x\sigma_1^y\sigma_0^y + \sigma_3^y\sigma_2^y\sigma_1^x\sigma_0^x), \end{aligned} \tag{4.9}$$

with coefficients c_i again derived from the molecular integrals. Under the Jordan-Wigner transformation all the qubits are used to store occupation numbers while in in the Bravyi-Kitaev transformation even qubits store occupations and odd qubits keep track of the parity of all the qubits with smaller indices. Hence, four qubits are needed to encode the ansatz state

$|\varphi_{\mathrm{HF}}\rangle = |0011\rangle$. All the required expectation values of the Jordan-Wigner H_2 Hamiltonian can be measured with five different sets of projective measurements.

Once the Hamiltonian is established, we need a variational ansatz to execute VQE. We choose to employ the Unitary Coupled Cluster (UCC) ansatz described in detail in Chapter 1. For H_2 in the minimal basis, the second quantized formulation of the UCC operator for single and double excitations corresponds to

$$U = \exp[\theta_{01}^{23}(a_2^\dagger a_3^\dagger a_1 a_0 - a_0^\dagger a_1^\dagger a_3 a_2)], \tag{4.10}$$

where θ_{01}^{23} is the coupled cluster amplitude that is variationally optimized. Note that in this case the single-excitation operators are effectively incorporated in the basis we are using (i.e. the single excitations rotate the basis and do not need to be applied explicitly in the circuit). Using the BK mapping this operator is expressed as follows:

$$\begin{aligned} U(\theta_{01}^{23}) = \exp\Bigg(i\frac{\theta_{01}^{23}}{8}\Big[&-\sigma_2^x\sigma_0^y + \sigma_2^y\sigma_0^x - \sigma_2^x\sigma_1^z\sigma_0^y + \sigma_2^y\sigma_1^z\sigma_0^x \\ &- \sigma_3^z\sigma_2^x\sigma_0^y + \sigma_3^z\sigma_2^y\sigma_0^x - \sigma_3^z\sigma_2^x\sigma_1^z\sigma_0^y + \sigma_3^z\sigma_2^y\sigma_1^z\sigma_0^x\Big]\Bigg). \end{aligned} \tag{4.11}$$

As the terms in $U(\theta_{01}^{23})$ only act with the identity and σ^z on qubits 1 and 3, the operator can be reduced to

$$\begin{aligned} U(\theta_{01}^{23}) &= \exp\left(i\frac{\theta_{01}^{23}}{2}[-\sigma_1^x\sigma_0^y + \sigma_1^y\sigma_0^x]\right) \\ &= \exp\left(-i\frac{\theta_{01}^{23}}{2}\sigma_1^x\sigma_0^y\right)\exp\left(i\frac{\theta_{01}^{23}}{2}\sigma_1^y\sigma_0^x\right), \end{aligned} \tag{4.12}$$

where we have relabeled qubit 2 as 1 and used the fact that the two operators commute. Finally, we observe that the two exponentials in $U(\theta_{01}^{23})$ perform the same operation when applied to the

reference state $|01\rangle$, explicitly:

$$\begin{aligned}
&\exp\left(i\frac{\theta_{01}^{23}}{2}\sigma_1^y\sigma_0^x\right)|01\rangle \\
&\quad=\left(\cos(\frac{\theta_{01}^{23}}{2})I+i\sin(\frac{\theta_{01}^{23}}{2})\sigma_1^y\sigma_0^x\right)|01\rangle \\
&\quad=\left(\cos(\frac{\theta_{01}^{23}}{2})I+i\sin(\frac{\theta_{01}^{23}}{2})\sigma_1^x\sigma_0^y\sigma_1^z\sigma_0^z\right)|01\rangle \\
&\quad=\left(\cos(\frac{\theta_{01}^{23}}{2})I-i\sin(\frac{\theta_{01}^{23}}{2})\sigma_1^x\sigma_0^y\right)|01\rangle \\
&\quad=\exp\left(-i\frac{\theta_{01}^{23}}{2}\sigma_1^x\sigma_0^y\right)|01\rangle. \qquad (4.13)
\end{aligned}$$

This allows us to define the ansatz via the $U(\theta_{01}^{23})$ operator simply as:

$$U(\theta_{01}^{23}) = \exp\left(-i\theta_{01}^{23}\sigma_1^x\sigma_0^y\right). \qquad (4.14)$$

Note that this form is only valid when the operator acts on the reference state $|01\rangle$ and when no other excitation operators are incorporated into the circuit. We also notice that this variational ansatz suffices to compute the exact ground state of H_2 in a minimal basis. In order to implement the UCC operator (4.14) above using Mølmer-Sørensen gates, we employ a technique first demonstrated in Müller et al.[238], formulae 10-12. If we consider arbitrary tensor products of qubit Pauli operators A and B with $[A,B] \neq 0$ we have:

$$\exp(-i\alpha A)\exp(i\theta B)\exp(i\alpha A) = \exp(i\theta B'), \qquad (4.15)$$

with $B' = \exp(-i\alpha A)B\exp(i\alpha A)$ and using the fact that Pauli operators are self-inverse:

$$B' = (\mathcal{I}\cos\alpha - iA\sin\alpha)B(\mathcal{I}\cos\alpha + iA\sin\alpha), \qquad (4.16)$$

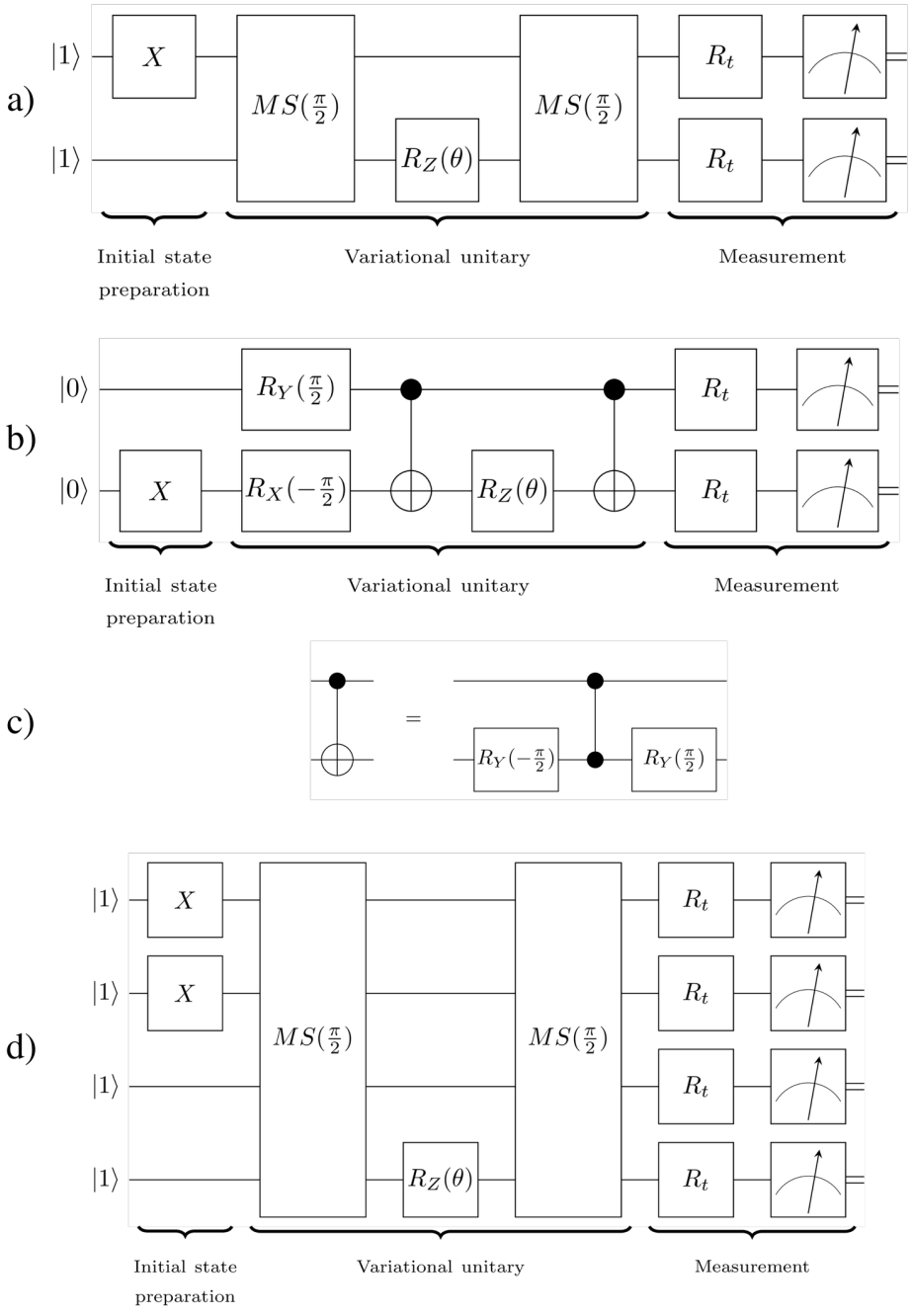


Figure 4.2: Circuits for VQE simulation of H_2 with the variational ansatz $\exp\left(-i\theta\sigma_1^x\sigma_0^y\right)$ applied on the initial state $|01\rangle$, corresponding the Hartree-Fock state: a) Implementation on TID with MS gates and the Bravyi-Kitaev encoding. b) Pre-compiled circuit using CNOT gates for the SQD architecture. c) Compilation of a CNOT gate to the native CZ_π for the X-mon device. d) Implementation for TID with MS gates and the JW encoding, implementing the variational unitary $\exp\left(-i\theta\sigma_3^x\sigma_2^x\sigma_1^x\sigma_0^y\right)$ action on the initial state $|0011\rangle$. The gate R_t represent the rotations employed to measure a specific string of Pauli matrices.

using the fact that A and B do not commute and therefore must anticommute we obtain:

$$B' = B\cos 2\alpha - \frac{i}{2}[A,B]\sin 2\alpha. \tag{4.17}$$

Specifically, for $\alpha = \pi/4$ and the case above

$$\begin{aligned}
\exp(-i\theta\sigma_y^1\sigma_x^j) &= \sin\left(i\theta\frac{i[\sigma_x^1\sigma_x^j,\sigma_z^1]}{2}\right) \\
&= \exp(-i\frac{\pi}{4}\sigma_x^1\sigma_x^j)\exp(-i\theta\sigma_z^1)\exp(i\frac{\pi}{4}\sigma_x^1\sigma_x^j) \\
&= \exp\left(i\frac{\pi}{4}\sigma_x^1\sigma_x^j\right)\exp\left(-i(\theta+\pi)\sigma_z^1\right)\exp\left(i\frac{\pi}{4}\sigma_x^1\sigma_x^j\right) \\
&= MS^{\{0,1\}}(\frac{\pi}{2},0)R_Z^1(\alpha)MS^{\{0,1\}}(\frac{\pi}{2},0).
\end{aligned} \tag{4.18}$$

Figure 4.2(a) depicts the final circuit for implementing VQE with this approximation. To achieve the same circuit in the SQD, we first derived the circuit using CNOT gates, following the method described in[243]. Figure 4.2(b) shows the pre-compiled circuit for the implementation on the SQD. The circuit expressed in CNOT gates can be compiled into the native CZ_π using the gate equivalence shown in 4.2(c). Finally, Figure 4.2(d) displays the circuit for implementing the same excitation operator in the JW representation using two global MS gates acting on four qubits.

4.2.2 LiH in a minimal basis

We obtained the molecular integrals for LiH at different internuclear separations using a Hartree-Fock calculation with an STO-6G basis set. For this particular application, we used the integrals in the natural orbital basis (NMO). NMOs are obtained by diagonalizing the exact one-electron reduced density matrix (1-RDM) of the system and are ordered by natural orbital occupation numbers (NOONs). It has been shown that NMOs with small NOONs or NOONs close to full-occupancy have a negligible effect in the electron correlation and therefore can be

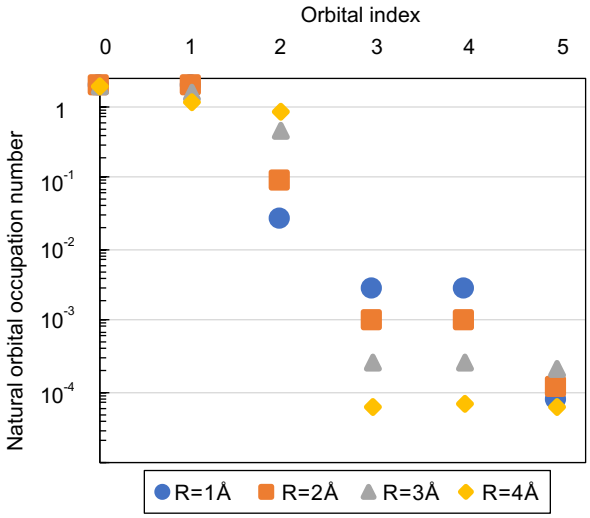


Figure 4.3: Natural orbital occupation numbers at four different internuclear separations for LiH, calculated at the CISD level. *Copyright (2018) by American Physical Society.*

discarded[85]. Consequently, approximate NMOs and NOONs, obtained from perturbation theory or truncated configuration interaction (CI) calculations, are usually employed to reduce the computational cost of more involved correlated calculations and for the selection of active spaces[345].

After the BK transformation, the Hamiltonian for LiH comprises 193 terms with amplitudes more significant than 10^{-10} Hartree. A VQE simulation using the UCC ansatz truncated to single and double excitations requires 12 qubits and involves 32 single-excitation operators and 168 double-excitation operators (without imposing spin constraints). To reduce the number of excitation operators in the calculation, we employed the NOONs derived from a configuration interaction calculation with single and double excitations (CISD) to select an appropriate active space. Fig. 4.3 shows the NOONs for the six molecular orbitals of LiH, calculated using CISD for four different internuclear separations. Based on the variations in the NOONs, we establish orbitals 1 to 4 as an appropriate active space.

A reasonable choice of excitation operators in the selected active space would be the singlet

double excitations from orbital 1 to orbitals 2, 3 and 4, respectively. Based on the NOONs, we expect the amplitude of the excitation operator from orbital 1 to orbital 2 to be largest. Similarly, we expect excitation operators from orbital 1 to orbitals 3 and 4 to have the same or similar amplitudes. Due to constraints in circuit depth, we consider only excitations from orbital 1 to orbitals 2 and 3, explicitly the operators: $a_5^\dagger a_4^\dagger a_3 a_2 - a_2^\dagger a_3^\dagger a_4 a_5$ and $a_7^\dagger a_6^\dagger a_3 a_2 - a_2^\dagger a_3^\dagger a_6 a_7$, where $a_i^\dagger(a_i)$ denote the creation (annihilation) operator in the i-th spin-orbital. In our notation, spin-orbitals with odd (even) indices correspond to spin-up (spin-down) electrons, with indices starting at 0.

Using the BK mapping, these operators can be expressed as:

$$\begin{aligned} a_5^\dagger a_4^\dagger a_3 a_2 - a_2^\dagger a_3^\dagger a_4 a_5 \equiv & \frac{i}{8}(\sigma_2^X \sigma_4^Y + \sigma_1^Z \sigma_2^X \sigma_3^Z \sigma_4^Y - \sigma_2^Y \sigma_4^X - \sigma_1^Z \sigma_2^Y \sigma_3^Z \sigma_4^X - \sigma_2^Y \sigma_4^X \sigma_5^Z \\ & - \sigma_1^Z \sigma_2^Y \sigma_3^Z \sigma_4^X \sigma_5^Z + \sigma_2^X \sigma_4^Y \sigma_5^Z + \sigma_1^Z \sigma_2^X \sigma_3^Z \sigma_4^Y \sigma_5^Z), \end{aligned} \tag{4.19}$$

and

$$\begin{aligned} a_7^\dagger a_6^\dagger a_3 a_2 - a_2^\dagger a_3^\dagger a_6 a_7 \equiv & \frac{i}{8}(\sigma_2^X \sigma_6^Y + \sigma_1^Z \sigma_2^X \sigma_3^Z \sigma_6^Y - \sigma_2^Y \sigma_6^X - \sigma_1^Z \sigma_2^Y \sigma_3^Z \sigma_6^X \\ & - \sigma_2^Y \sigma_3^Z \sigma_5^Z \sigma_6^X \sigma_7^Z - \sigma_1^Z \sigma_2^Y \sigma_5^Z \sigma_6^X \sigma_7^Z + \sigma_2^X \sigma_3^Z \sigma_5^Z \sigma_6^Y \sigma_7^Z + \sigma_1^Z \sigma_2^X \sigma_5^Z \sigma_6^Y \sigma_7^Z). \end{aligned} \tag{4.20}$$

The initial state of the simulation, corresponding to the Hartree-Fock wavefunction, is the state $|000000000101\rangle$, which simplifies to $|000001\rangle$ in the active space required for the selected excitation operators. A full-simulation of LiH with the two double excitation operators listed above would require at most 32 MS gates for a single Trotter step if all the subterms were going to be implemented. To make the simulation affordable in the current device we approximated each of the operators using only the first subterm, corresponding to $\sigma_2^X \sigma_4^Y$ and $\sigma_2^X \sigma_6^Y$, respectively. The

absolute error in the total energy introduced by this approximation is smaller than chemical accuracy within the basis set used, when compared to the full configuration interaction (FCI) reference solution. The two selected subterms can be implemented using MS gates and single qubit rotations as follows:

$$\exp[-i\alpha\sigma_2^X\sigma_4^Y] \equiv MS^{\{2,4\}}(\frac{\pi}{2},0)R_z^2(\alpha)MS^{\{2,4\}}(-\frac{\pi}{2},0), \tag{4.21}$$

and

$$\exp[-i\beta\sigma_2^X\sigma_6^Y] \equiv MS^{\{2,6\}}(\frac{\pi}{2},0)R_z^2(\beta)MS^{\{2,6\}}(-\frac{\pi}{2},0). \tag{4.22}$$

As the entangling operations involve only qubits 2, 4 and 6, we can efficiently construct an effective Hamiltonian involving only operations on these qubits. The corresponding 3-qubit Hamiltonian has the form:

$$\begin{aligned} H =& c_0 I + c_1 Z_0 + c_2 Z_1 + c_3 Z_2 + c_4 Z_1 Z_0 + c_5 Z_2 Z_0 + c_6 Z_2 Z_1 + c_7 X_1 X_0 \\ &+ c_8 Y_1 Y_0 + c_9 X_2 X_0 + c_{10} Y_2 Y_0 + c_{11} X_2 X_1 + c_{12} Y_2 Y_1. \end{aligned} \tag{4.23}$$

As described in [142], this reduction is related to the previously observed fact that BK computational basis states naturally reflect certain spin symmetries, in some cases allowing for a more compact representation than the corresponding JW mapping. The expectation value of the final Hamiltonian can be measured by performing three rounds of projective measurements at each point in the parameter space.

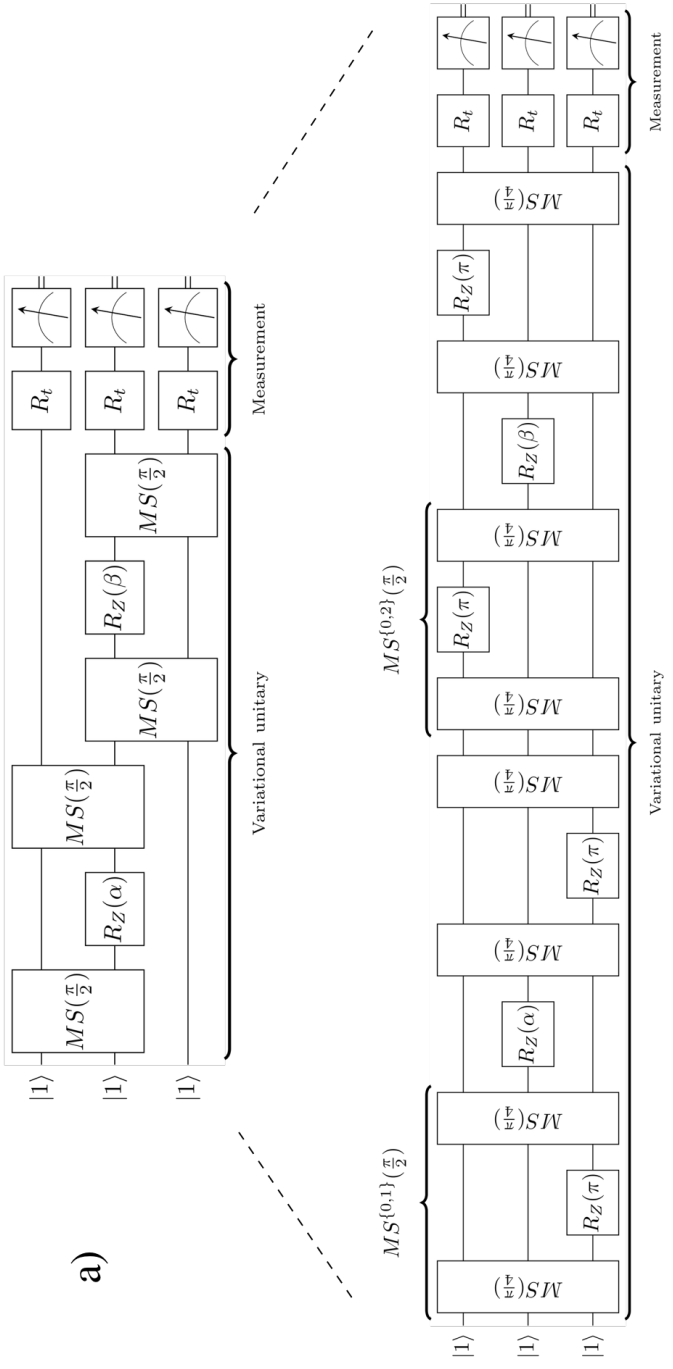


Figure 4.4: Implementation of the VQE simulation of LiH on trapped ion quantum computers with the variational ansatz of Eq. (4.21): a) Pre-compiled circuit with MS gates and the BK encoding. b) Actual circuit implemented on the TID, compiled using a refocusing technique with half-entangling MS gates. The gates R_t represent the rotations employed to measure a specific string of Pauli matrices.

Figure 4.4(a) depicts the circuit for implementing VQE for LiH with the ansatz and initial states described above. We chose to implement it in the experiment using the circuit shown in Figure 4.4(b), based on a refocusing technique[238]. Here, an addressed π-phase shift between two half-entangling MS gates decouples the addressed qubit from the two remaining qubits, effectively implementing an entangling operation only among the target qubits. The initial state for the calculation is rotated to $|111\rangle$, which corresponds to the lowest energy state of the qubit register for the TID architecture.

4.3 Experimental results

4.3.1 H_2 simulations

Given the simplicity of our variational ansatz for H_2, which contains a single variational parameter θ, we performed exploration of the parameter space by executing the variational circuit and estimating the expectation values of the terms of the Hamiltonian for several values of θ. In the SQD experiment, we scanned 1000 points in the interval $[-\pi, \pi]$. Each expectation value was estimated using 10000 samples per point. In the FID experiment, we scanned 100 points in the interval $[0, 2\pi]$, and each expectation value was determined using 100 repetitions per data point. In terms of absolute times, operations in the superconducting qubit architecture are approximately three orders of magnitudes faster than operations with trapped ions, which facilitates executing experiments with more circuit repetitions.

Figure 4.5 shows the expectation values obtained from the SQD and TID experiments on the BK Hamiltonian for H_2. We observe that expectation values follow the correct patterns but fluctuate as a result of the noise sources affecting the device. A particularly informative aspect of the plot is the expectation value of the Z_0Z_1 operator, that happens to be a stabilizer of the state prepared by the chosen variational ansatz. As a result, the expectation value of this operator is constant with

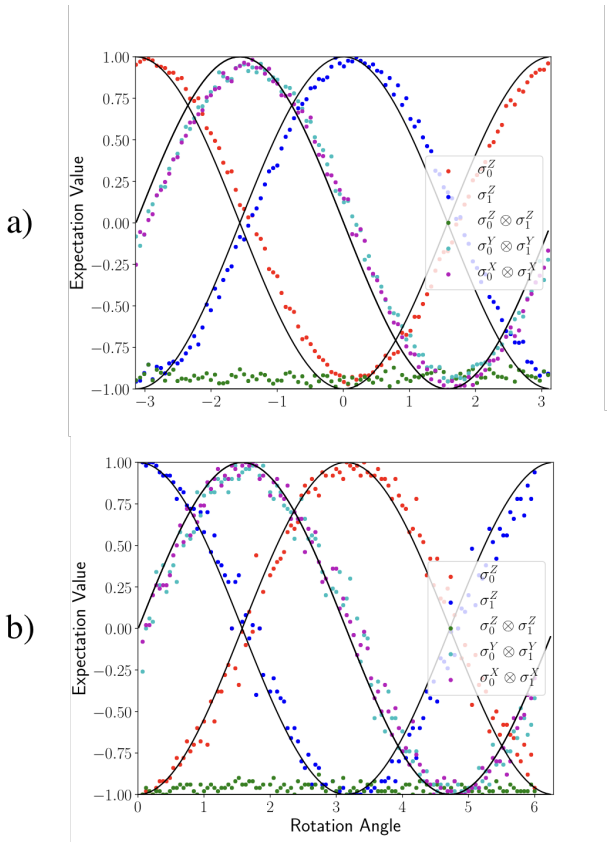


Figure 4.5: Expectation values of the Hamiltonian terms for H_2 in two qubits, using the BK mapping as measured in a) SQD and b) TID experiments for different values of the variational parameter θ. The solid black lines nearest to the data indicate the theoretical values. Values were estimated using a 100 repetitions per point in the TID experiment and 1000 repetitions per point in the SQD experiment. Each plot presents only 100 equally spaced points clarity.

respect to *theta*, with an expectation value of -1.0. Any deviation from this expectation value is a result of errors that do not commute with the unitary used as ansatz, in addition to QPN error. In both experiments, we observe a slight deviation from the ideal behavior confirming the presence of such non-commutative spurious terms in the evolutions implemented by the devices. The results for both devices also present a systematic shift in the experimental expectation values towards the right compared to the theoretical values. The lowest statistical error of the SQD experiments allows us to observe this pattern more clearly.

From the scans of the expectation values it is possible to compute the potential energy surface by

using the expression:

$$E(R,\theta) = \sum_i h_i(R) \exp O_i(\theta), \tag{4.24}$$

where we have made explicit the dependence of the Hamiltonian coefficients on the molecular coordinates (R) and the dependence of the expectation values on the variational parameters, θ. If the scan contains a sufficiently large number of points, it is possible to carry out the VQE optimization at each internuclear distance, R, by directly selecting the minimum energy in the grid. We can also apply statistical techniques such as Gaussian processes regression or a simple sinusoidal fit to model either the expectation values or the potential energy surface itself to estimate the value of these properties at any point in the parameter space. To evaluate the overall performance of the quantum simulator, we can compare the direct measurements or their fitted functions against classical numerical simulations, i.e. full diagonalization of the Hamiltonian. We notice that full diagonalization is not scalable but affordable for the small systems studied here and we only use it as a reference for comparison and not as part of the VQE implementation.

Given the number of grid points employed in the SQD experiment, we opted for directly computing the minimum energy at each internuclear distance from the grid. Error bars were calculated with a GPC fitting procedure to remove systematic statistical biases in the data and to reduce the statistical uncertainty at each point. Figure 4.6 presents the result of this procedure. Specifically, Figure 4.6(a) compares the PES of the ground state of H_2 estimated on the quantum computer with the theoretical PES. The energies computed with the VQE procedure are higher, as expected from the variational nature of the method. As noticed earlier, the variational ansatz used for H_2 suffices to obtain the ground state in a minimal basis and therefore the discrepancies observed between the experimental and theoretical values can be associated to errors in the quantum computer.

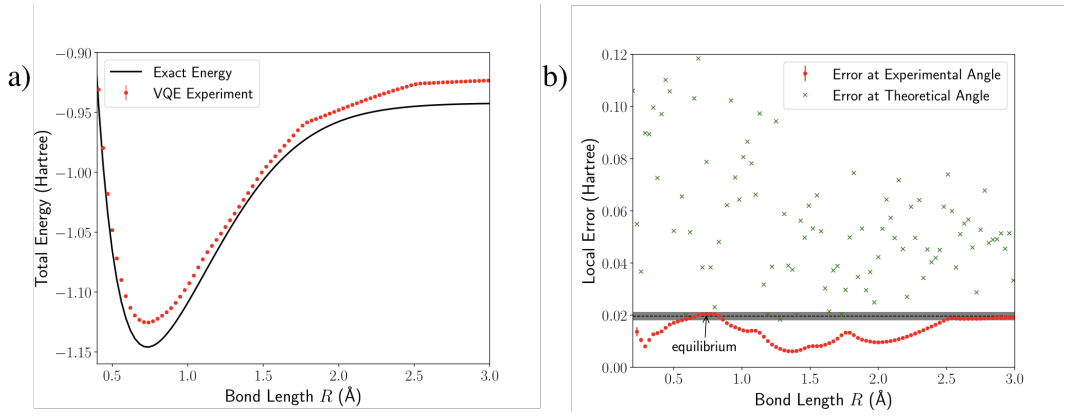


Figure 4.6: Results of the SQD experiment: a) PES of molecular hydrogen in a minimal basis obtained in the experiment, compared with the exact theoretical value. b) Errors in VQE energy surface. Red dots show error in the experimentally determined energies. Green dots correspond to errors in the energies that would have been obtained experimentally by running the circuit at the theoretical optimal value of θ instead of the optimal θ from the experiment. The discrepancy between green and red dots provides experimental evidence for the robustness of VQE against certain experimental errors. The gray band encloses the chemically accurate region relative to the experimental energy of the atomized molecule. The dissociation energy is calculated with respect to the energy at the equilibrium geometry, indicated by the arrow.

4.6(b) displays the absolute difference between the theoretical energy and the optimal variational energy along the PES, showing that the latter is considerably above the quantum chemistry threshold (≈ 1.5 mHartree). However, errors have similar magnitudes at the equilibrium point and at the dissociation limit, which allow us to estimate the dissociation energy of H_2 with an error of $(8 \pm 5) \times 10^{-4}$ mHartree compared to the theoretical result. This result constitutes the first estimation of a molecular property within chemical accuracy using a quantum computer, reported in[248]. 4.6(b) also compares the errors in the energies that would have been obtained experimentally by running the circuit at the theoretical optimal value of θ instead of the optimal θ from the experiment. The significant difference between these values and the actual experimental error is an indication of the robustness of the VQE procedure against specific errors of the quantum device. In this particular case, this difference can be partly associated with the systematic shift in the expectation values observed in Figure 4.5(a).

For the TID experiments, we implemented VQE for H_2 for the two-qubit BK Hamiltonian and the four-qubit JW Hamiltonian as well. The initial states for these simulations are $|01\rangle$ and $|0011\rangle$,

respectively. When the corresponding variational unitaries act upon these states, as shown in Figures 4.2(a) and 4.2(d), the entangled states generated belong to a decoherence-free subspace (DFS) protected against correlated dephasing - a significant source of error in trapped ion qubit devices [142]. To investigate the impact of this decoherence channel in the implementation, we performed simulations with initial states outside this DFS. To implement these simulations we applied a unitary rotation of the Hamiltonian that effectively changes the signs of the Hamiltonian coefficients, $h_i(R)$. For the BK implementation, this accounts to using the state $|11\rangle$ as the initial state while maintaining the same unitary for state preparation and performing the energy calculation with the modified Hamiltonian.

The results of the experiments described above can be seen in Figure 4.7(a) and 4.7(b). The experiments include two simulations for the BK Hamiltonian, the first corresponding to a DFS-protected initial state ($|01\rangle$) and the second corresponding to an unprotected ($|11\rangle$) one. Both experiments were executed with MS gates of the same fidelity $\in$99(3)% [‡]. The remaining experiments correspond to simulations for the JW Hamiltonian for H_2 executed with MS gates with fidelities 97(3) and 93(3) respectively. 4.7(a) shows the PES calculated for all these experiments from the parameter scan while 4.7(b) compares the magnitude of the dissociation energy for H_2 (displayed as binding energy). The results showed an increasing error with decreasing fidelity and increased correlated dephasing error. Interestingly, the unprotected BK experiment with MS fidelity of 99(3)% has similar accuracy to the protected JW experiment with MS fidelity of 97(3)%, which is a strong indication of the effect of DFS protection, especially taking into account that the JW experiment involves twice as many qubits as the BK one.

[‡]MS gate entanglement fidelity was estimated from population averages and the parity contrast relating to the coherence of the respective Bell or GHZ state generated from $|11\rangle$ or $|1111\rangle$ at the conclusion of the operation [34,199,284].

Figure 4.7: a) Potential energy curves of the molecular hydrogen ground state. The black line corresponds to the theoretical value calculated in the chosen minimal basis. All other lines are derived from weighted sinusoidal fits to the energy surfaces formed from the experimentally obtained expectation values. The data sets vary the number of qubits, the Hartree Fock input states, encodings and gate fidelities as listed in the legend below the figures. **b)** Data from panel a) normalized to the theoretical dissociation energy at large internuclear separations R. The dashed and dotted lines indicate the well depth associated with the binding energy of the molecule and the position of the energy minimum, respectively. **c)** VQE implementation. The BK $\mathrm{HF} = |01\rangle$ parameter scan fit result is shown as experimental reference with its 1σ confidence band. Points with error bars indicate the five VQE runs performed. The inset shows the last iterations of one particular run that failed to converge to the target value (blue line) with experimental data depicted by red and a noise-free circuit simulation by black symbols. *Copyright (2018) by American Physical Society.*

Finally, we implemented the full VQE algorithm at five different internuclear separations R as described in Figure 4.1, yielding the results shown in Figure 4.7(c). The optimization procedure is performed by passing the measurement results of the quantum circuit to a Nelder-Mead simplex algorithm[242] running on a classical computer. The optimization is initialized by choosing a random initial value of θ_0, and the energy is calculated at each iteration according to Equation 4.24. In parallel, we execute a noise-free simulation of the circuit at each iteration to monitor the convergence towards the theoretically expected energy.

Generally, the algorithm converges in simulation and experiment. Figure 4.8(a) illustrates the convergence process, while Figure 4.8(b) compares the experimental and theoretical results of the optimization process. The observed statistical fluctuations are a result of the QPN errors and noisy gate operations described in Subsection 4.2.0.2. To determine the optimal value from the Nelder-Mead iterations, we implemented a sinusoidal fit to the 1D parameter space of the energy explored throughout all iterations, with each point weighted according to the QPN contributions from its constituent expectation value measurements, as described in[142]. Figure 4.7(c) shows each run's result superimposed on the previously discussed parameter scan. Error bars for the VQE points are derived from the above fitting procedure. In most cases, the full VQE optimization converges to a value close to the energy estimated from the grid scan (Figure 4.7(a)).

4.3.1.1 Decoherence simulation for H_2 experiment in TID

To understand the effect of various decoherence channels, we performed a simulation of the entire circuit of the two-qubit H_2 experiment using the open source framework OpenFermion[225] and in-house code to simulate the noise channels. We simulated the experiments based on the actual experimental times.

We assume that throughout the execution of the entire state preparation circuit, the qubits experience dephasing, e.g. due to magnetic field fluctuations induced by the environment. We

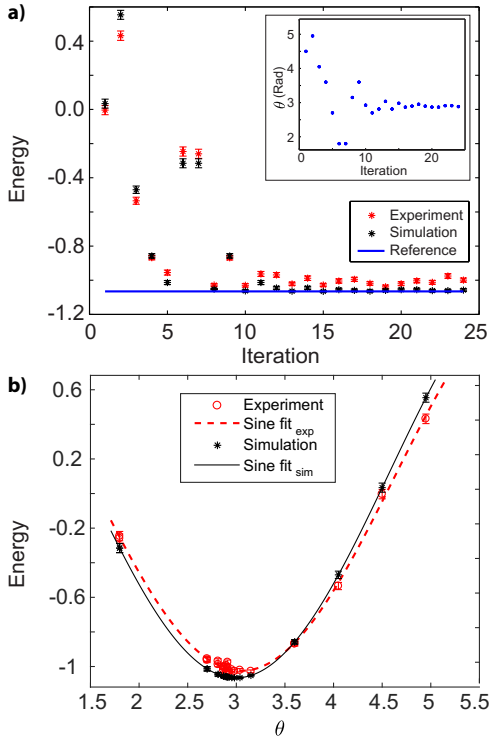


Figure 4.8: Online VQE runs for $R = 0.5$ Å. **a)** Energy vs. Iteration number (target = blue line, simulated result = black, experimental result = red) Inset: Rotation angle α vs. Iteration number. Error bars derived from quantum projection noise. **b)** Energy vs. Rotation angle α with sinusoidal fitting (simulation = black, experiment = red). *Copyright (2018) by American Physical Society.*

model the dephasing via an i.i.d. channel of the form

$$\varepsilon(\rho) = \varepsilon_{i_1} \circ \varepsilon_{i_2}(\rho)$$

where

$$\varepsilon_i(\rho) = (1 - p_d)\rho + p_d \sigma_i^z \rho \sigma_i^z,$$

is a Kraus map and p_d is the probability for a single phase flip. In our simulations, we applied the dephasing channel to all the qubits after the application of each gate in the circuit of Figure 4.2(a), with probability $p_d = 1.0 - \exp(-T_g/T_2)$, where T_g is the physical time of the gate and T_2 is the dephasing time. We employed $T_2 = 40$ ms, as determined from Ramsey experiments on a single

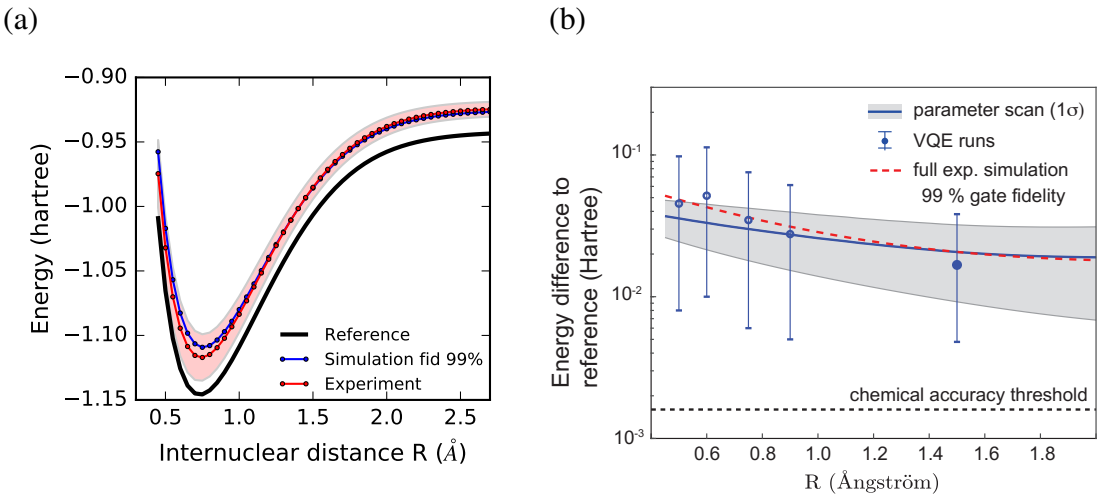


Figure 4.9: a) Simulation of decoherence channels in H_2 under the BK mapping. The plot compares the simulated energy curves for different values of the fidelity of the MS gate and the experimental results. The reference curve corresponds to the exact diagonalization of the Hamiltonian. The decoherence channels include one-qubit dephasing acting on all the qubits during the state preparation and two-qubit depolarizing errors in the MS gate. b) Energy errors of the reconstructed H_2 potential energy surface and the influence of decoherence. A full configuration interaction (FCI) calculation performed in the chosen molecular basis serves as reference value. The red line corresponds to a full simulation of the quantum circuit, including multiple decoherence channels and the experimentally determined gate fidelity. *Copyright (2018) by American Physical Society.*

ion. In addition to dephasing, we model the effect of errors in the MS gates using a two-qubit depolarizing channel.

Here, we consider all single- and two-qubit errors with the same probability. For a 2-qubit MS gate, the noise is described by the quantum operation

$$\begin{aligned}\varepsilon_{MS}(\rho) =&(1-p_{MS})\rho + \frac{p_{MS}}{15}\sum_{i\in\Lambda_a}\sum_{\alpha\in\Lambda_\alpha}\sigma_i^\alpha\rho\sigma_i^\alpha \\ &+\frac{p_{MS}}{15}\sum_{j_1,j_2\in\Lambda_a}\sum_{\alpha,\beta\in\Lambda_\alpha}\sigma_{j_1}^\alpha\sigma_{j_2}^\beta\rho\sigma_{j_1}^\alpha\sigma_{j_2}^\beta,\end{aligned}$$

where p_{MS} is the probability of a MS depolarizing error, and Λ_a and Λ_α correspond to the set of indexes for the active ions and Pauli matrices, respectively. For the 2-qubit MS gate, there are 15 possible Pauli errors (6 single-qubit and 9 two-qubit), resulting in the prefactor 1/15. The probability p_{MS} is related to the fidelity of the gate as $F = 1 - \frac{14}{15}p_{MS}$[37]. The experimental fidelity

estimated for the 2-qubit MS gate is 0.99. In the simulation, the two-qubit depolarizing channel is applied with probability p_{MS} after the MS gate.

Figure 4.9 display the results of the simulation for H_2 under the BK mapping. Our simulation appears to account for the observed experimental errors along with a significant portion of the energy curve. We observe an uneven upshift of the energy values that effectively reduces the estimated well depth (dissociation energy) estimate. While our simulation is close in magnitude to the observed results, we note that other factors such as faulty measurement operators (related to basis rotations and detection fidelity) could also contribute to the discrepancies.

4.3.2 LiH SIMULATION ON A MULTIQUBIT TRAPPED-ION QUANTUM COMPUTER

We performed the experiment for LiH only on TID. We started by performing a parameter scan for the variational ansatz depicted in Figure 4.4(a). We scan the expectation values for the operations in the Hamiltonian of Equation 4.23 sequentially in a grid determined by the intervals $\alpha = [1.5, 6]$ and $\beta = [2, 5]$. Each term in the Hamiltonian was estimated using 500 circuit runs. We combined these results using Equation 4.24 to compute the energy landscape for each internuclear separation R. An example for $R = 1.6\,$Åis shown in Figure 4.10(a) along with a section of the parameter space estimated from theory. To compute the PES from the experimental data, we applied two different fitting procedures: (1) a two-dimensional quadratic fit to the energy minimum and (2) a Gaussian process regression (GPR) git. From the fitted function, we calculate the minimum energy at each internuclear separation, yielding the results shown in Figure 4.10(b). We observe that the GPR results lie above the theoretical PES while the quadratic fit lies below, illustrating the sensibility of the results to the statistical treatment.

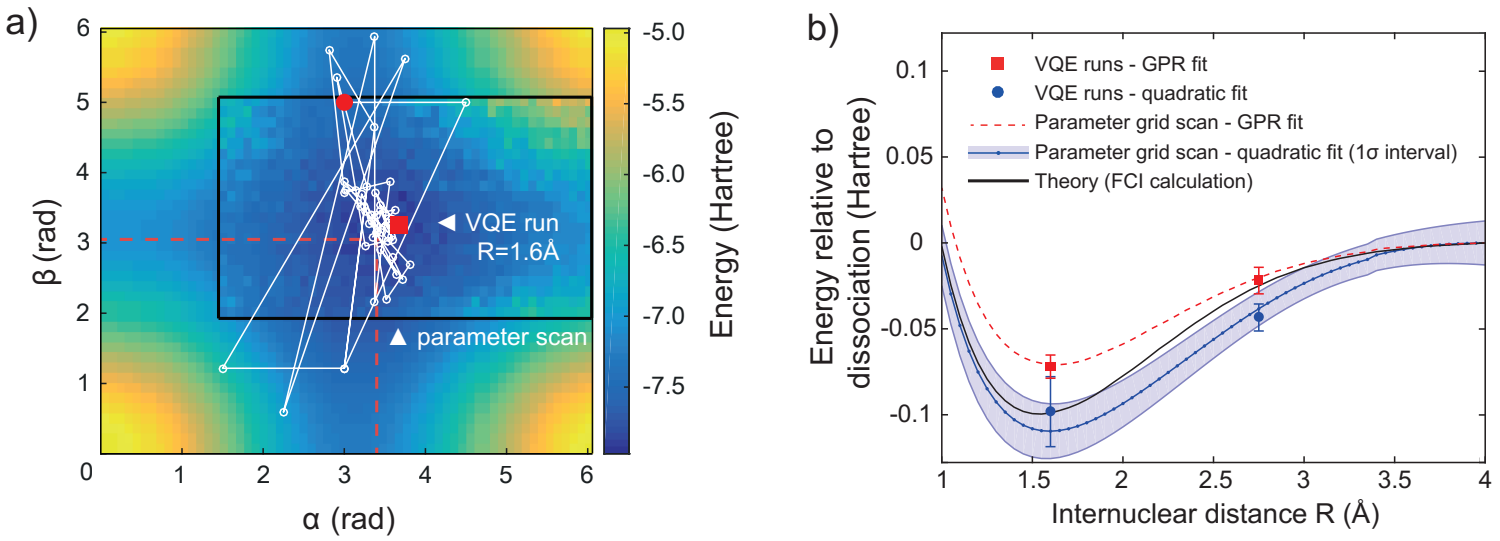


Figure 4.10: LiH results. **a)** Energy landscape at internuclear distance $R = 1.6$Å . The experimentally scanned parameter range, indicated by the black box inset, is superimposed on a theoretical calculation. The dashed red lines mark the coordinates of the targeted energy minimum. Connected white lines show the steps taken by the VQE algorithm, with the starting point marked by a filled red dot and the terminus by a red square. **b)** The theoretical LiH potential energy surface calculated for the minimal basis set (black) is shown in comparison with experimentally obtained results, offset to overlap at maximum distance R to better illustrate the well-depth differences in the grid scan reconstructions. In absolute values, all measured data is above the FCI calculation. The data points result from sampling the energy landscape $\rangle H(R) \langle (\alpha, \beta)$ using the VQE algorithm or the parameter scan shown in a) and fitting the explored space with a Gaussian process regression (GPR) based machine learning algorithm (red dashed line) or a 2D quadratic fit (blue solid line). We computed the error bars from the fits with the underlying data weighted by quantum projection noise. The slight kink close to $R = 3.5$Å is due to the interplay of rounding errors introduced in the fitting routine with small deviations originating in our active space approximation. *Copyright (2018) by American Physical Society.*

We also implemented the full VQE algorithm for LiH. As described in detail in Ref.[142], the use of the bare Nelder-Mead optimization in this calculation usually fails due to statistical fluctuations that stops the algorithm before reaching the true minimum. To address this issue, we switch the optimization to a hybrid algorithm[346] that also incorporates an element of simulated annealing by introducing random perturbations, sampled from a distribution D and added to the cost function in Equation 4.24. These perturbations force the VQE algorithm to continuously sample the surroundings of the minimum as shown in Figure 4.10(a) without converging any further. The minimum's location is then estimated from a fitting procedure applied to the sampled points, as described in Ref.[142]. The distribution D is chosen to be a uniform distribution with a range in the same order as the magnitude of the QPN. Once the fluctuations in the VQE algorithm become of the same order as the perturbations, the algorithm samples 10-20 additional points using the procedure described above. The energy is then calculated using a quadratic or a GPR fitting procedure, yielding the results shown in Figure 4.10(b). We observe that the results are consistent with the baseline PES obtained from the scan and highlight the discrepancy between the two fitted functions. It is likely that we can reduce this discrepancy by taking a larger number of measurements per data point at the expense of increasing the execution time of the algorithm.

4.3.3 Discussion

The VQE experiments presented above in both SQD and TID illustrate one of the remarkable features of this algorithm: its resilience against certain types of errors in pre-error corrected quantum devices. These *variationally suppressible* errors, as characterized in Ref.[226], correspond to errors that can be mitigated by changes of the variational parameters and therefore are naturally suppressed by the classical optimization loop incorporated into VQE, i.e., calibration errors in some of the gates, such as the one observed in Figure 4.5 for the H_2 implementation.

Despite this resilience, the VQE algorithm relies on expectation values and therefore the

precision to which operators are measured have a significant impact on the quality of the results, as illustrated by the challenges faced in the LiH simulations. Furthermore, this makes VQE particularly sensitive to errors in the measurement step. Apart from the improvement that can be done at the level of the physical implementation in order to enhance measurements[227], a couple of algorithmic improvements are also possible: for example, one might incorporate a second optimization loop to variationally correct the rotations employed in the measurement step, as described in[142]. Another improvement is the incorporation of post-selection techniques that discard measurements that violate known symmetries of the Hamiltonian as described in Ref.[205,221,226].

4.4 Conclusions

In this chapter, we summarized the first scalable demonstrations of quantum simulation of Chemistry using superconducting qubits[248] and multi-qubit trapped ion[142] quantum devices. We demonstrated the application of the VQE algorithm to compute the PES of the H_2 and LiH molecules, employing a minimal basis set and both BK and JW representations of the Hamiltonians. Our implementation employs a variational ansatz inspired by unitary coupled cluster (UCC) and takes advantage of the particular symmetries of the molecules to simplify the number of qubits and circuit depths compared to the regular UCC approach. Our simulation achieved the first calculation of a molecular property within chemical accuracy using a quantum computer and offered the first experimental evidence of noise resilience of the VQE algorithm against experimental errors. Our results also illustrate some of the challenges of implementing VQE on NISQ devices. In particular, we showed the instabilities caused by different error sources in the optimization process and proposed methods to circumvent them.

Our experiments also illustrate the importance of incorporating domain knowledge from quantum chemistry and experimental quantum computing in the implementation of the VQE

algorithm. In particular, we show the utility of standard quantum chemistry techniques, such as active space selection, in VQE. This strategy could be exploited in VQE calculations for much larger systems where only the more strongly correlated orbitals of a molecule are modeled on the quantum computer, leaving the weakly correlated part to efficient classical techniques such as density functional theory[53] and perturbation theory[141]. This type of calculation could be accomplished using a variation of the hybrid quantum-classical scheme employed for VQE (Figure 4.1). The knowledge of the natural noise processes of the quantum platforms is also crucial in the VQE implementation. An example of this is the use of DFS for protecting the calculation against correlated decoherence noise in trapped ions.

Since the realization of these experiments, quantum simulation with variational methods has experienced tremendous progress in both theoretical and experimental sides. On the theory front, the introduction of new variational ansatzes with more favorable circuit depth[78,179], more efficient representations of the quantum chemistry Hamiltonian[19], and general techniques for error mitigation on NISQ devices[98,169,331], promise to reduce the implementation cost of VQE and increase the accuracy of the calculations. On the experimental front, the race for achieving the first demonstration of quantum advantage is delivering ever growing quantum chips with lower error rates[51,113,233].

Along with this experimental progress, the field has experienced significant improvements in the infrastructure for accessing and controlling the devices. Nowadays, a variety of cloud-based platforms[69,270,313,322] offer software for quick prototyping and deployment of quantum algorithms on NISQ computers, providing a test bed for all the recent algorithm improvements. Along with this, libraries for manipulating quantum chemistry Hamiltonians and implementing specific variational ansatzes for Chemistry have also been consolidated[225,270]. We hope that the progress observed in the last couple of years will bring us closer to the simulation of molecules with tens of

spin orbitals (qubits) soon.

Part II

New variational algorithms for quantum computing and statistical modeling

This page intentionally left blank.

5

Quantum autoencoders for efficient compression of quantum data

Apart from minor modifications, this chapter originally appeared as[279]:

5.1 Introduction

Quantum technologies, ranging from quantum computing to quantum cryptography, have been found to have increasingly powerful applications for a modern society. Quantum simulators for chemistry, for example, have been recently shown to be capable of efficiently calculating molecular energies for small systems[248]; the capability for larger scale simulations promises to have deep implications for materials design, pharmacological research, and an array of other potentially life-changing functions[14]. A limiting factor for nearly all of these applications, however, is the amount of quantum resources that can be realized in an experiment. Therefore, for experiments now and in the near future, any tool which can reduce the experimental overhead in terms of these resources is especially valuable.

For classical data processing, machine learning via an autoencoder is one such tool for dimensional reduction[38,207,208], as well as having application in generative data models[118]. A

classical autoencoder is a function whose parameters are optimized across a training set of data which, given an $(n+k)$-bit input string x, attempts to reproduce x. Part of the specification of the circuit, however, is to erase some k bits during the process. If an autoencoder is successfully trained to reproduce x at the output at least approximately, then the remaining n bits immediately after the erasure (referred to as the *latent space*) represent a compressed encoding of the string x. Thus, the circuit "learns" to encode information that is close to the training set.

In this chapter, we introduce the concept of a quantum autoencoder which is inspired by this design for an input of $n+k$ qubits. Because quantum mechanics is able to generate patterns with properties (e.g. superposition and entanglement) that is beyond classical physics, a quantum computer should also be able to *recognize* patterns that are beyond the capabilities of classical machine learning. Thus, the motivation for a *quantum* autoencoder is simple; such a model allows us to perform analogous machine learning tasks for quantum systems without exponentially costly classical memory, for instance, in dimension reduction of quantum data. A related work proposing a quantum autoencoder model establishes a formal connection between classical and quantum feedforward neural networks where a particular setting of parameters in the quantum network reduces to a classical neural network exactly[350]. In this work, we provide a simpler model which we believe more easily captures the essence of an autoencoder, and apply it to ground states of the Hubbard model and molecular Hamiltonians.

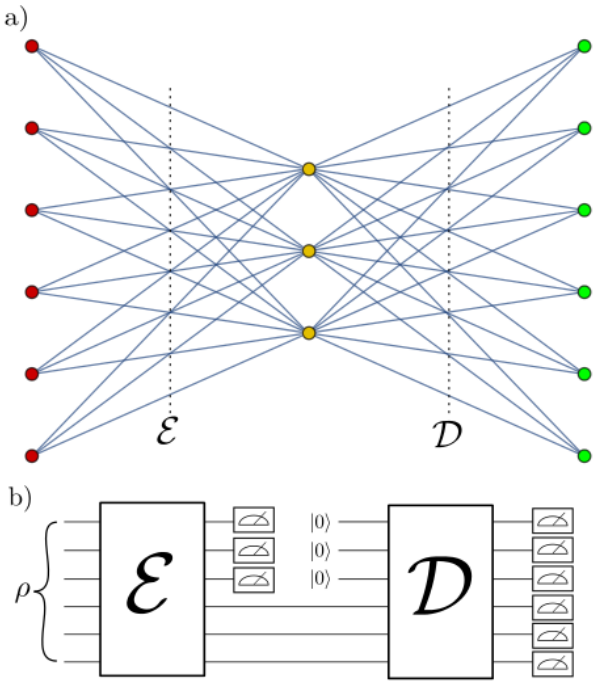


Figure 5.1: a) A graphical representation of a 6-bit autoencoder with a 3-bit latent space. The map $\mathcal{E}$ encodes a 6-bit input (red dots) into a 3-bit intermediate state (yellow dots), after which the decoder $\mathcal{D}$ attempts to reconstruct the input bits at the output (green dots). b) Circuit implementation of a 6-3-6 quantum autoencoder.

5.2 Quantum Autoencoder Model

In analogy with the model of classical autoencoders, the quantum network has a graphical representation consisting of an interconnected group of nodes. In the graph of the quantum network, each node represents a qubit, with the first layer of the network representing the input register and the last layer representing the output register. In our representation, the edges connecting adjacent layers represent a unitary transformation from one layer to the next. Autoencoders, in particular, shrink the space between the first and second layer, as depicted in Figure 5.1a.

For a quantum circuit to embody an autoencoder network, the information contained in some of the input nodes must be discarded after the initial “encoding" $\mathcal{E}$. We imagine this takes place by

tracing over the qubits representing these nodes (in Figure 5.1b, this is represented by a measurement on those qubits). Fresh qubits (initialized to some reference state) are then prepared and used to implement the final “decoding" evolution $\mathcal{D}$, which is then compared to the initial state.

The learning task for a quantum autoencoder is to find unitaries which preserve the quantum information of the input through the smaller intermediate latent space. To this end, it is important to quantify the deviation from the initial input state, $|\psi_i\rangle$, to the output, ρ_i^{out}. Here, we will use the expected fidelity[361] $F(|\psi_i\rangle, \rho_i^{out}) = \langle\psi_i| \rho_i^{out} |\psi_i\rangle$. We thus describe a successful autoencoding as one in which $F(|\psi_i\rangle, \rho_i^{out}) \approx 1$ for all the input states.

A more formal description of a quantum autoencoder follows: Let $\{p_i, |\psi_i\rangle_{AB}\}$ be an ensemble of pure states on $n+k$ qubits, where subsystems A and B are comprised of n and k qubits, respectively. Let $\{U^{\vec{p}}\}$ be a family of unitary operators acting on $n+k$ qubits, where $\vec{p} = \{\mathrm{p}_1, \mathrm{p}_2, \dots\}$ is some set of real parameters defining a unitary quantum circuit. Also let $|a\rangle_{B'}$ be some fixed pure reference state of k qubits. Using classical learning methods, we wish to find the unitary $U^{\vec{p}}$ which maximizes the average fidelity, which we define to be the cost function,

$$C_1(\vec{p}) = \sum_i p_i \cdot F(|\psi_i\rangle, \rho_{i,\vec{p}}^{out}), \tag{5.1}$$

where,

$$\rho_{i,\vec{p}}^{out} = (U^{\vec{p}})^{\dagger}_{AB'} \operatorname{Tr}_B \left[U^{\vec{p}}_{AB} \left[\psi_{i_{AB}} \otimes a_{B'} \right] (U^{\vec{p}}_{AB})^{\dagger} \right] (U^{\vec{p}})_{AB'} , \tag{5.2}$$

and we have abbreviated $|\psi_i\rangle\langle\psi_i|_{AB} = \psi_{i_{AB}}$ and $|a\rangle\langle a|_{B'} = a_{B'}$. Equivalently, the goal is to find the best unitary $U^{\vec{p}}$ which, on average, best preserves the input state of the circuit in Figure 5.2 where instead of tracing over the B system, we employ a swap gate and trace over the B' system.

If we denote $U|\psi_i\rangle = |\psi_i'\rangle$, then it is clear that that we are comparing two systems in the primed basis that differ only by a swap on the B and B' systems. More precisely, the global state after the

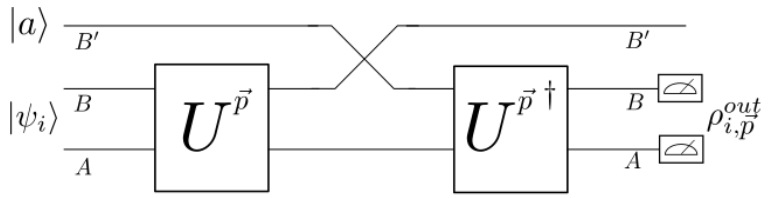


Figure 5.2: A quantum autoencoder circuit. The goal is to find $\vec{p}$ such that the averaged $F(|\psi\rangle_i, \rho_{i,\vec{p}}^{out})$ is maximized.

swap is $|\psi_i'\rangle_{AB'} \otimes |a\rangle_B$. It is then compared via the fidelity to $|\psi_i'\rangle_{AB}$, and the fidelities in the cost function C_1 arise by tracing out over B',

$$F(|\psi_i'\rangle_{AB}, \mathrm{Tr}_{B'}[\psi'_{i_{AB'}} \otimes a_B]) = F(|\psi_i'\rangle_{AB}, \rho'_{i_A} \otimes a_B), \tag{5.3}$$

where $\rho'_{i_A} = \mathrm{Tr}_{B'}\left[|\psi_i'\rangle\langle\psi_i'|_{AB'}\right]$. The proof that the fidelities in Eq. (5.1) and Eq. (5.3) are equal is shown explicitly in Appendix 5.6.1. Note that both sides of the fidelity in Eq. (5.3) involve the subscript i; this is a result of considering a swap test specific to an input state that is indexed with i. Such knowledge about the input states may be unrealistic in some settings. To get around this, we might consider instead tracing over the AB system and computing the fidelity of the "trash system" of B' compared to the reference state,

$$F(|a\rangle_{B'}, \mathrm{Tr}_{AB}\left[\psi'_{i_{AB'}} \otimes a_B\right]) = F(|a\rangle_{B'}, \rho'_{i_{B'}}), \tag{5.4}$$

where $\rho'_{i_{B'}} = \mathrm{Tr}_A[|\psi_i'\rangle\langle\psi_i'|_{AB'}]$. We henceforth refer to $\rho'_{i_{B'}}$ as the "trash state" of the circuit. Note that this now simplifies the matter of employing a swap test, since it is no longer conditioned on the input's index. It is straightforward to see in the above circuit that perfect fidelity (i.e. $C_1 = 1$)

can be achieved by a unitary U if and only if, for all i:

$$U|\psi_i\rangle_{AB} = |\psi_i^c\rangle_A \otimes |a\rangle_B \,. \tag{5.5}$$

where $|\psi_i^c\rangle_A$ is some compressed version of $|\psi_i\rangle$. This follows because, if the B and B' systems are identical when the swap occurs, the entire circuit reduces to the identity map. However, this occurs precisely when the trash state is equal to the reference state, i.e., $F(|a\rangle_{B'}, \rho'_{i_{B'}}) = 1$. This implies that it is possible to accomplish the learning task of finding the ideal $U^{\vec{p}}$ by training *only on the trash state*. Furthermore, because Eq. (5.5) is completely independent of $U^\dagger$, this suggests that the circuit of Figure 5.2 can be reduced further. We then consider an alternative definition of the cost function in terms of the trash state fidelity,

$$C_2(\vec{p}) = \sum_i p_i \cdot F(\mathrm{Tr}_A \left[U^{\vec{p}} |\psi_i\rangle \langle\psi_i|_{AB} (U^{\vec{p}})^\dagger\right], |a\rangle_B), \tag{5.6}$$

Note, however, that the cost functions of Eq. (5.1) and Eq. (5.6) are not in general the same (in fact, $C_1 \leq C_2$). However, in practice, one must consider resource limitations; it is not hard to see that preparing copies of a fixed reference state would be easier than requiring identical copies of the input state to use in a SWAP test on the entire output state. For some applications of a quantum autoencoder, it may also be the case that one has limited access to or limited knowledge of the input state.

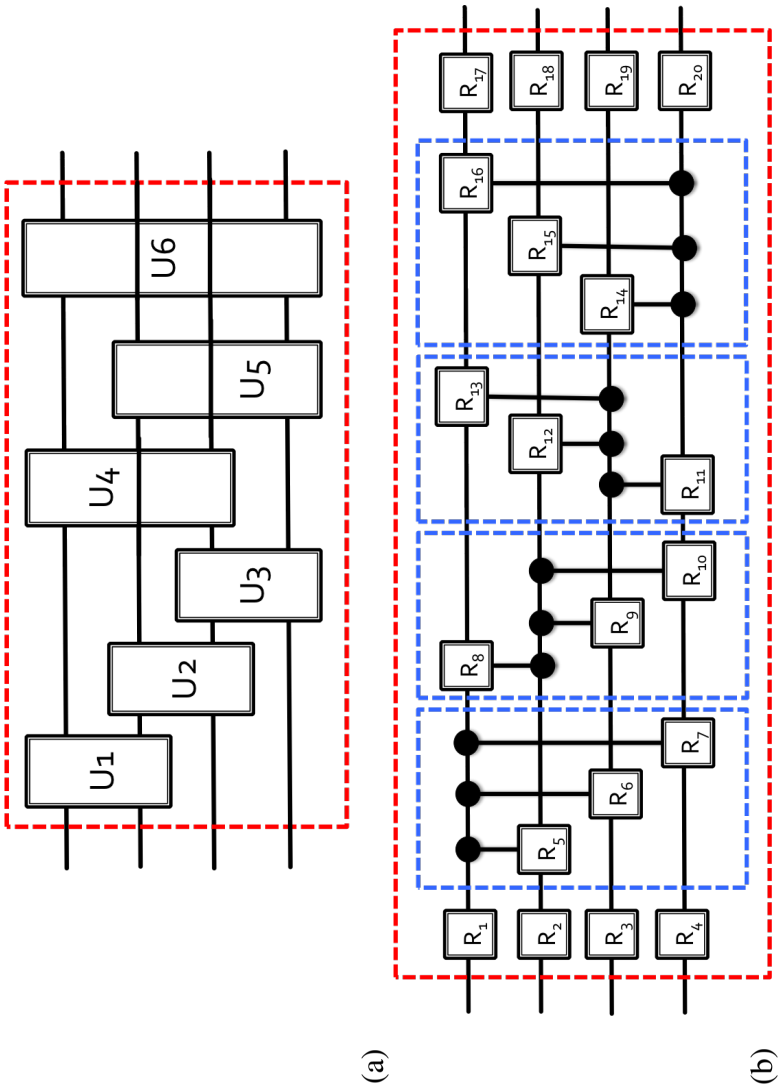


Figure 5.3: Two programmable circuits employed as autoencoder models: a) Circuit A: a network of all the possible two-qubit gates (denoted by U_i) among the qubits. b) Circuit B: a network comprising all the possible controlled general single-qubit rotations (denoted by R_i) in a qubit set, plus single qubit rotations at the beginning and at the end of the unit-cell. All the circuits are depicted in the case of a four-qubit input. The unit-cell is delimited by the red dotted line.

It is interesting to note that, if we only care about circuits where $C_2 \approx 1$, we can re-imagine the problem to being one of finding a particular disentangling. It has been shown that, employing a circuit of exponential depth, one is always able to perform a disentangling operation[305], but to perform this operation in constant or polynomial depth is hard, and so classical heuristics are often used to find quantum circuits that are as close to optimal as possible. Also, information-theoretic bounds have been explored in this context before, both in the context of one-shot compression and one-shot decoupling[84,96]. In particular, it may be possible to apply the decoupling methods used in the "mother of all protocols"[3]. However, because the heuristics involved in choosing efficient-to-implement families of unitaries are largely ad-hoc, it is difficult to say if these bounds are meaningful in the context of a quantum autoencoder.

5.3 Implementation of the quantum autoencoder model

To implement the quantum autoencoder model on a quantum computer we must define the form of the unitary, $U^{\vec{p}}$ (Eq. (5.2)) and decompose it into a quantum circuit suitable for optimization. For the implementation to be efficient, the number of parameters and the number of gates in the circuit should scale polynomially with the number of input qubits. This requirement immediately eliminates the possibility of using a $(n+k)$-qubit general unitary as $U^{\vec{p}}$ due to the exponential scaling in the number of parameters needed to generate them.

One alternative for the generation of $U^{\vec{p}}$ is to employ a programmable quantum circuit[83,319]. This type of circuit construction consists of a fixed networks of gates, where a polynomial number of parameters associated to the gates i.e. rotation angles, constitute $\vec{p}$. The pattern defining the network of gates is regarded as a unit-cell. This unit-cell can ideally be repeated to increase the flexibility of the model. For the numerical assessment presented in this work, we employed two simple programmable circuits illustrated in Figure 5.3.

Circuit A has a unit-cell comprising a network of general two-qubit gates where we have considered all the possible pairings between qubits, as illustrated in Figure 5.3a for the four-qubit case. Accordingly, this model requires $15n(n-1)/2$ training parameters per unit-cell. Recalling that an arbitrary n-qubit unitary requires $2^{2n}-1$ real parameter to specify, we see that for $n=4$, one unit cell of our circuit is far from universal, requiring only 90 real parameters compared to 255. A network of arbitrary two qubit gates can be easily implemented using state of the art superconducting qubit technologies[23] and the standard decomposition of a two-qubit gate into three CNOT gates and single-qubit rotations[186]. Arbitrary two qubit-gates have been also implemented using ion traps[135] and quantum dots[342].

Circuit B has a unit-cell comprising all the possible controlled one-qubit rotations among a set of qubits, complemented with a set of single qubit rotations at the beginning and at the end of the unit-cell, as shown in Figure 5.3b for the four-qubit case. We start considering the rotations controlled by the first qubit, followed by the rotations controlled by the second qubit and so on. Accordingly, our second model comprises $3n(n-1)+6n$ (totaling 60 for $n=4$) training parameters per unit-cell and can be implemented in state of the art quantum hardware using the standard decomposition of controlled unitaries into two CNOT gates and single-qubit rotations[243]. This model is also general and can be modified by adding constraints to the parameters. For instance, one could consider the initial and final layers of rotations to be all the same.

Once the circuit model has been chosen, we must train the network by maximizing the autoencoder cost function Eq. (5.6), in close analogy to classical autoencoders. Our training procedure adopts a quantum-classical hybrid scheme in which the state preparation and measurement are performed on the quantum computer while the optimization is realized via an optimization algorithm running on a classical computer. Such hybrid schemes have been proposed in the context of quantum machine learning[20,114] and variational algorithms for quantum

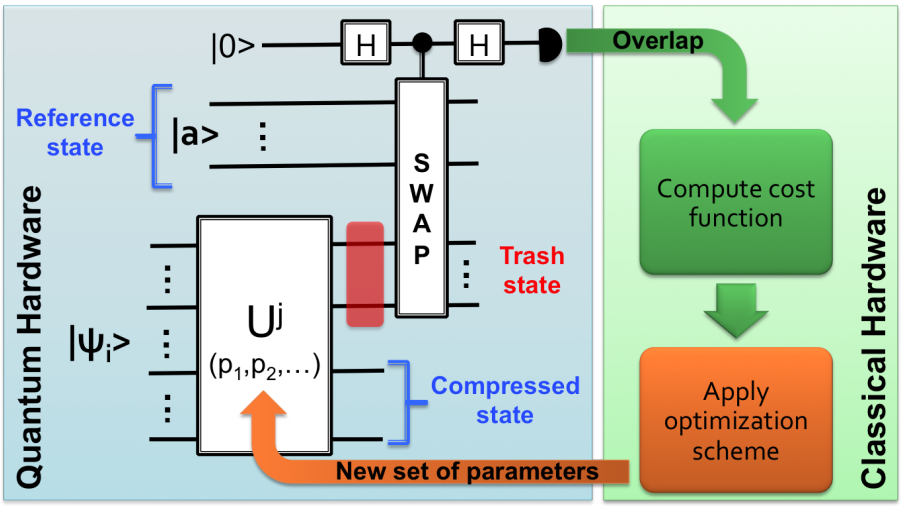


Figure 5.4: Schematic representation of the hybrid scheme for training a quantum autoencoder. After preparation of the input state, $|\psi_i\rangle$, the state is compressed by the application of the parameterized unitary, $U^{\vec{p}}$. The overlap between the reference state and the trash state produced by the compression is measured via a SWAP test. The results for all the states in the training set are collected to compute the cost function that is minimized using a classical optimization algorithm. The process is repeated until achieving convergence on the cost function and/or the values of the parameters, $\vec{p} = (p_1, p_2, \dots)$.

simulation[202,226,255,353]. In the later case, several experimental demonstrations have been successfully carried out[248,255,287].

As described in Section 5.2, the cost function of the quantum autoencoder is defined as the weighted average of fidelities between the trash state produced by the compression, and the reference state. These fidelities can be measured via a SWAP test[52] between the reference state and the trash state. Accordingly, our quantum register must comprise the input state, $|\psi_i\rangle$, and the reference state. In a single iteration of our training algorithm, we perform the following steps for each of the states in the training set:

1. Prepare the input state, $|\psi_i\rangle$, and the reference state. We assume these preparations to be efficient.

2. Evolve under the encoding unitary, $U^{\vec{p}}$, where $\vec{p}$ is the set of parameters at a given optimization step.

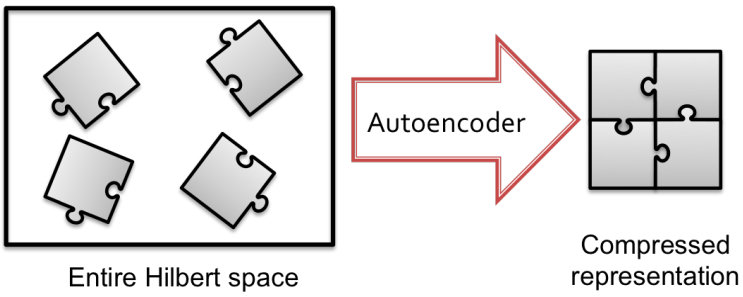


Figure 5.5: Graphical representation of Hilbert space compression. Given that the states of interest have support on only a subset $\mathcal{S}$ of the Hilbert space (gray pieces), the quantum autoencoder finds an encoding that uses a space of size $|\mathcal{S}|$.

3. Measure the fidelity between the trash state and the reference state via a SWAP test.

With the estimates of all the fidelities, the cost function (Eq. (5.6)) is computed and fed into a classical optimization routine that returns a new set of parameters for our compression circuit. These steps are repeated until the optimization algorithm converges. Given that the value of the cost function is upper bounded by 1, we performed the optimization by minimizing the value of the function $\log_{10}(1-C_2)$. This procedure is widely used in machine learning applications and helps prevent numerical instabilities[39]. A graphical summary of the hybrid scheme for training a quantum autoencoder is shown in Figure 5.4.

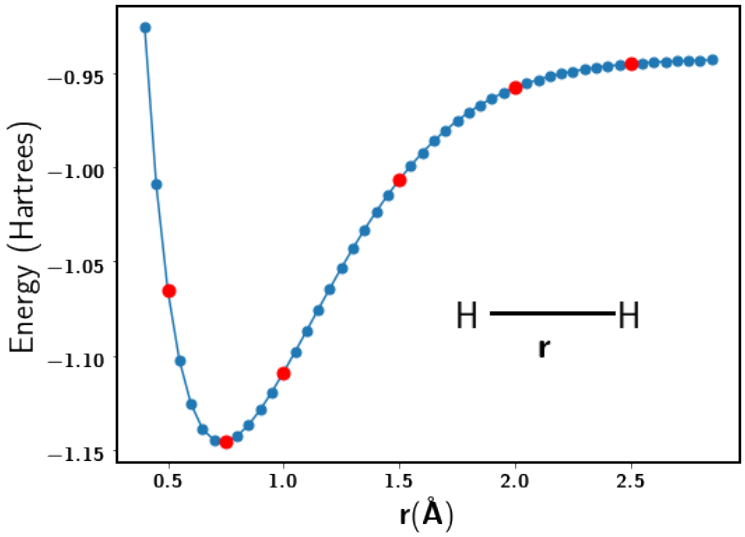


Figure 5.6: Potential energy surface for the hydrogen molecule using a STO-6G basis set. The ground states at the red dots where used as training set for the quantum autoencoder. The ground states at the blue dots were used for testing.

5.4 Application to quantum simulation

Consider a set of states, $\{|\psi_i\rangle\}$, with support on a subset of a Hilbert space $\mathcal{S} \subset \mathcal{H}$. Using a quantum autoencoder, we could find an encoding scheme that employs only $\log_2 |\mathcal{S}|$ qubits to represent the states instead of $\log_2 |\mathcal{H}|$, with a trash state of size $\log_2 |\mathcal{H} - \mathcal{S}|$. This idea is graphically depicted in Figure 5.5. This situation is usually encountered for eigenstates of many-body systems due to special symmetries.

Fermionic wavefunctions, for instance, are eigenfunctions of the particle number operator, same as the fermionic state vectors. Consequently, an eigenstate of a system with η particles is spanned exclusively by the subspace of fermionic state vectors with the same number of particles[141], that has size $\binom{N}{\eta}$ with N the number of fermionic modes. This result has direct implications for the design of quantum algorithms for simulation, suggesting that the number of qubits required to store fermionic wavefunctions could be reduced up to $\log \binom{N}{\eta}$ if an appropriate mapping can be found. The same situation is encountered for the spin projection operator, thus reducing the size of

the subspace spanning a specific fermionic wavefunction even further.

Generally, the number of particles of the system is part of the input when finding eigenstates of many-body systems. In quantum chemistry simulations, the spin projection of the target state is also known. Many classical algorithms for simulating quantum systems take advantage of these constraints to reduce their computational cost[141]. However, the standard transformations employed to map fermionic systems to qubits, namely the Bravyi-Kitaev (BK) and the Jordan-Wigner (JW) mappings[300,336], do not exploit these symmetries and thus employ more qubits than formally needed.

In this scenario, a quantum autoencoder could be trained to compress fermionic wavefunctions obtained using a quantum simulation algorithm that has been run using the standard transformations. The compression schemes obtained through this procedure could be employed to reduce the use of quantum memory, if the wavefunction needs to be stored. It also could save quantum resources for the simulation of systems with similar symmetries. To illustrate this idea, we simulated a quantum autoencoder applied to molecular wavefunctions.

Within the Born-Oppenheimer approximation, the non-relativistic molecular Hamiltonian can be written as

$$H = h_{nuc} + \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s \tag{5.7}$$

where h_{nuc} corresponds to the classical electrostatic repulsion between nuclei, and the constants h_{pq} and h_{pqrs} correspond to the one- and two-electron integrals (see Section 2.2). The operators $a_p^\dagger$ and a_p creates and annihilates an electron in the spin-orbital p. After applying either the JW or the BK transformation, the molecular Hamiltonian can be expressed as $H = \sum_i^M c_i H_i$, with M scaling as $O(N^4)$. In this case, the operators H_i correspond to tensor products of Pauli matrices and the real coefficient c_i are linear combinations of the one- and two-electron integrals. For a fixed set of

nuclei and a given number of electrons, the molecular integrals as well as the coefficients c_i are functions of the internal coordinates of the molecule, $\vec{R}$.

For instance, consider the Hamiltonian of molecular hydrogen in the STO-6G minimal basis set[141]. Using the JW transformation, the corresponding Hamiltonian acting on four qubits adopts the generic form[300]:

$$\begin{aligned} H = {} & c_0 I + c_1(Z_0 + Z_1) + c_2(Z_2 + Z_3) + c_3 Z_0 Z_1 + \\ & c_4(Z_0 Z_2 + Z_1 Z_3) + c_5(Z_1 Z_2 + Z_0 Z_3) + c_6 Z_2 Z_3 \\ & + c_7(Y_0 X_1 X_2 Y_3 - X_0 X_1 Y_2 Y_3 - Y_0 Y_1 X_2 X_3 + X_0 Y_1 Y_2 X_3) \end{aligned} \tag{5.8}$$

In this case, the coefficients c_i are a function of the internuclear distance, r. By solving the Schrödinger equation for the Hamiltonians at different values of r, we can obtain the ground state energy for molecular hydrogen and construct the potential energy surface (PES) shown in Figure 5.6. We expect that the ground state wavefunctions along the PES conserve the same number of particles and projection spin symmetries, turning this set of states into an excellent target for compression.

To illustrate the previous idea, we classically simulated a quantum autoencoder taking six ground states of the hydrogen molecule at different values of r, $\{|\Psi(r_i)\rangle\}_{i=1}^{6}$, as our training set. In this case, the weights of the states are chosen to be all equal. In real applications, we can imagine that the ground states are obtained using a quantum algorithm such as the quantum variational eigensolver[255]. We trained the circuit model described in Figure 5.3 to compress the training set of four-qubit states to two qubits and to one qubit, using $|0\rangle^{\otimes 2}$ and $|0\rangle^{\otimes 3}$ as reference states, respectively. Once the circuits were trained we tested them on 44 ground states corresponding to values of r different from those of the training set. This selection of the training and testing sets is shown in Figure 5.6.

Table 5.1: Average fidelity ($\mathcal{F}$) error after one cycle of compression and decompression using the quantum autoencoder trained from ground states of the Hydrogen molecule. We also report the error in the energy of the decoded state. (Maximum and minimum errors shown within parenthesis). 6 states were used for training and 44 more were used for testing. These results were obtained with the L-BFGS-B optimization.

Circuit	Final size (# qubits)	Set	$-\log_{10}(1-\mathcal{F})$ MAE	-$\log_{10}$ Energy MAE (Hartrees)
Model A	2	Training	6.96(6.82-7.17)	6.64(6.27-7.06)
	2	Testing	6.99(6.81-7.21)	6.76(6.18-7.10)
	1	Training	6.92(6.80-7.07)	6.60(6.23-7.05)
	1	Testing	6.96(6.77-7.08)	6.72(6.15-7.05)
Model B	2	Training	6.11(5.94-6.21)	6.00(5.78-6.21)
	2	Testing	6.07(5.91-6.21)	6.03(5.70-6.21)
	1	Training	3.95(3.53-5.24)	3.74(3.38-4.57)
	1	Testing	3.81(3.50-5.38)	3.62(3.35-4.65)

* MAE: Mean Absolute Error. Log chemical accuracy in Hartrees ≈-2.80

The classical simulation was performed using a Python script supplemented with the QuTiP library[161,162]. To simulate general two-qubit gates we employed the decomposition described in Ref.[135]. Arbitrary single-qubit rotations were implemented by decomposing them into Pauli-Z and Pauli-Y rotations, $R = R_z(\theta_1)R_y(\theta_2)R_z(\theta_3)$, ignoring global phases[243]. The optimization was performed using the SciPy implementation of the Basin-Hopping (BS) algorithm[347]. We also employed the L-BFGS-B method[55] with a numerical gradient (central finite difference formula with step size $h = 10^{-8}$). In the optimization of both circuit models, the parameters were constrained to the range $[0, 4\pi)$. The optimization of each circuit was initialized by setting the parameters to randomly selected values.

Table 5.1 shows the average error in the fidelities and the energies obtained after a cycle of compression and decompression through the optimal quantum autoencoder. We observe that both circuit models are able to achieve high fidelities for the encoding, producing decoded wavefunctions with energies that are close to the original values within chemical accuracy (1kcal/mol $\equiv 1.6 \times 10^{-3}$ Hartrees $\equiv 43.4$ meV). This accuracy requirement assures that quantum chemistry predictions have enough quality to be used for the estimation of thermochemical

Table 5.2: Average fidelity ($\mathcal{F}$) error after one cycle of compression and decompression using the quantum autoencoder trained from ground states of the Hydrogen molecule. We also report the error in the energy of the decoded state. (Maximum and minimum errors shown within parenthesis). 6 states were used for training and 44 more were used for testing. These results were obtained with the L-BFGS-B optimization.

Circuit	Final size (# qubits)	Set	$1-F$ MAE	Energy MAE (Hartrees)
Model	2	Training	$< 10^{-6}$	$< 10^{-6}$
A	2	Testing	$< 10^{-6}$	$< 10^{-6}$
	1	Training	$< 10^{-6}$	$< 10^{-6}$
++	1	Testing	$< 10^{-6}$	$< 10^{-6}$
Model	2	Training	$< 10^{-6}$	$< 10^{-6}$
B	2	Testing	$< 10^{-6}$	$< 10^{-6}$
	1	Training	$< 10^{-3}$	$< 10^{-3}$
	1	Testing	$< 10^{-3}$	$< 10^{-3}$

* MAE: Mean Absolute Error. Chemical accuracy in Hartrees $\approx$0.0016

properties such as reaction rates[257].

Figure 5.7 illustrates the optimization process of a quantum autoencoder. We compared two different optimization algorithms, L-BFGS-B and Basin-Hopping. The parameters were initialized at random and the same guess was employed for both optimizations. As observed in Figure 5.7, both algorithms required a similar number of cost function evaluations to achieve similar precision and exhibit a monotonic reduction of the difference between the cost function and its ideal value with the number of function evaluations. The implementation of the quantum autoencoder in state of the art architectures can benefit from the use of algorithms that do not require gradient evaluations and have a larger tolerance to the presence of noise in the hardware, such as Basin-Hopping.

To gain insight into the compression process, we plotted the density matrices of the compressed states and compared them with the density matrix of the original state in Figure 5.8, for three different values of r. The sparsity of the original input density matrix is due to the symmetry of the Hamiltonian for molecular hydrogen, whose eigenvectors have support on only 2 computational basis states, allowing for a compression up to a single qubit. Although the quantum autoencoder

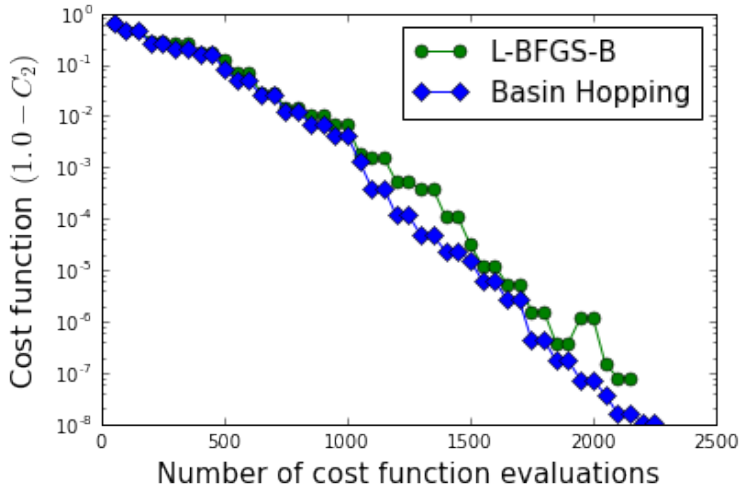


Figure 5.7: A plot of the cost function versus the number of cost function evaluations during the training process. This example corresponds to a quantum autoencoder for compression of wavefunction of H_2 from 4 to 2 qubits using circuit A with a training set of six ground states. We compared the L-BFGS-B and the Basin-Hopping algorithms for optimization.

achieves high fidelity with both types of circuit, the structure of the density matrices indicates that the forms of the compressed space and therefore the forms of the encoding unitaries differ between the two circuit models. As the values of r change, the relation between the features of the input space, here represented by the elements of the density matrix, and the features of the compressed space become apparent.

In addition to the example of molecular hydrogen, we tested the autoencoder compression scheme with ground states of the Hubbard model and the H_4 molecule. We considered half-filled Hubbard models with 2-sites and 4-sites arranged in a two-leg ladder (2×1 and 2×2 lattices, respectively). The Hamiltonian for these systems is given by

$$H = -t \sum_{<i,j>} \sum_{\sigma} a^{\dagger}_{i,\theta} a_{j,\sigma} + U \sum_{i} a^{\dagger}_{i,\uparrow} a_{i,\uparrow} a^{\dagger}_{i,\downarrow} a_{i,\downarrow} \tag{5.9}$$

where $a^{\dagger}_{i,\theta}$ and $a_{i,\sigma}$ create and annihilate an electron at site i with spin σ, respectively. The summation in the first term runs over all the interacting sites, denoted as $< i, j >$. We used periodic

Figure 5.8: Visualization of the input space and the latent (compressed) spaces for three different testing instances of the H_2 compression, corresponding to three different bond distances, r. The input and latent spaces are characterized as the density matrices of the input and compressed states. Letters (A) and (B) denote the type of circuit employed to construct the encoding unitary. The size of the register (in qubits) appears within parenthesis. Integer labels starting at 1 denote the computational basis states in ascending order from $|00\cdots0\rangle$ to $|11\cdots1\rangle$.

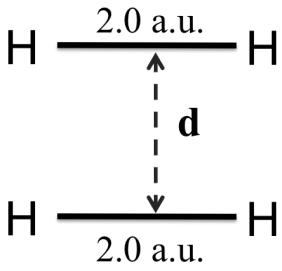


Figure 5.9: H_4 molecule in a parallel configuration, with the hydrogen atoms forming a rectangle. We obtained the ground states of this system at different values of d, with the bond distance in the two hydrogen molecules fixed to 2 atomic units (a.u.).

boundary conditions along the horizontal direction and open boundary conditions in the vertical direction. As in the case of molecular Hamiltonians, Hubbard Hamiltonians can be mapped to qubits using either the JW or the BK transformation, requiring two qubits per site.

We trained the two circuits of Figure 5.3 to compress the ground states of the Hubbard Hamiltonians obtained by setting t to six different values equally spaced between 0.9 and 1.1, with $U = 2.0$. The optimization process was repeated three times starting at randomly selected values. The same procedure was applied to the ground states of the H_4 system at six different values of the bond distance d (0.6, 1.4, 2.2, 3.0, 3.8 and 4.6 atomic units) for the geometry described in Figure 5.9.

Table 5.3 shows the lowest errors obtained for the compression of the Hubbard models and the H_4 system. Errors are quantified as the difference between the final value of the cost function (Eq. (5.6)) and the ideal value of 1. Recall that the cost function corresponds to the average fidelity over the training set. We observe that the ground states of the two-sites Hubbard model can be compressed from 4 to 3 qubits using both circuit types. However, only circuit B is able to compress these these states from 4 to 2 qubits and 4 to 1 qubits with an error below 10^{-3}. The same circuit-type achieves an error smaller than 10^{-4} when compressing the ground states of the H_4 system from 8 to 7 qubits. In contrast, circuit A is unable to obtain errors below 10^{-3} for H_4. In the case of the 4-sites Hubbard model, none of the circuit models proposed here was able to obtain

Table 5.3: Final error in the cost function (Eq. (5.6)) obtained after training a quantum autoencoder for compression of the ground states of two-sites and four-sites Hubbard models and the H_4 molecule along a parallel path. Six ground states were used for training each system. These results were obtained with the L-BFGS-B optimization.

Circuit	System	Compression rate* $n_o \to n_l$	$-\log_{10}(1-C_2)$
Model A	Hubbard 2x1 sites	$4 \to 3$	7.52
		$4 \to 2$	1.15
		$4 \to 1$	1.13
	Hubbard 2x2 sites	$8 \to 7$	2.28
		$8 \to 6$	1.42
	H_4	$8 \to 7$	1.53
		$8 \to 6$	1.6
Model B	Hubbard 2x1 sites	$4 \to 3$	6.82
		$4 \to 2$	3.92
		$4 \to 1$	4.02
	Hubbard 2x2 sites	$8 \to 7$	2.29
		$8 \to 6$	2.31
	H_4	$8 \to 7$	4.33
		$8 \to 6$	1.15

* n_o and n_l stand for the number of qubits in the original register and the number of qubits in the latent space, respectively.

errors below 10^{-3}.

The differences between the performances of the two circuit models described above exemplifies how the ansatz employed for the autoencoder unitary impacts the degree of compression achievable with the autoencoder model. Compression of a particular set of states could be achieved more easily with a dedicated ansatz designed for that purpose. One form of unitary that can serve as a template to design such dedicated ansatzes is given by the expression

$$U(\vec{\alpha}) = e^{-i\sum_i \alpha_i H_i} \tag{5.10}$$

where the real numbers α_i are the parameters for optimization and the terms H_i are local interactions consisting of tensor products of Pauli matrices. The experimental implementation of Eq. (5.10) would benefit from the techniques developed for quantum simulation algorithms[117].

Another interesting direction is to employ circuits for the preparation of tensor network states[293,294,299] as the autoencoder ansatz. Such circuits could provide a systematic way of adapting the circuit complexity to particular applications by changing parameters such as the bond dimension.

Finally, we point out that the maximum rate of lossless compression achievable with a quantum autoencoder is predetermined by the size of the subspace spanning the training set. Consequently, a given set of states might only admit a small or null compression rate. For instance, consider a fermionic system with 8 fermionic modes and 4 particles, such as a half-filled 4-sites Hubbard model or the H_4 molecule in a minimal basis set studied here. Based solely on the constrain in the number of particles, these 8-qubits systems could be compressed to $\log_2 \binom{8}{4} \approx 7$ qubits. Compression beyond this point could be achieved if an extra symmetry constraint is present. In general, we expect fermionic systems where the number of fermionic modes is considerably larger than the number of particles to be good candidates for compression.

5.5 Discussion

We have introduced a general model for a quantum autoencoder – a quantum circuit augmented with learning via classical optimization – and have shown that it is capable of learning a unitary circuit which can facilitate compression of quantum data, particularly in the context of quantum simulations. We imagine that the model can have other applications, such as compression protocols for quantum communication, error-correcting circuits, or perhaps to solve particular problems directly. A natural application for quantum autoencoders is state preparation. Once a quantum autoencoder has been trained to compress a specific set of states, the decompression unitary ($U^\dagger$) can be used to generate states similar to those originally used for training. This is achieved by preparing a state of the form $|\Psi_I\rangle \otimes |a\rangle$ and evolving it under $U^\dagger$, where $|\Psi_I\rangle$ has the

size of the latent space and $|a\rangle$ is the reference state used for training.

Autoencoders as state preparation tools have direct application in the context of quantum variational algorithms[202,226,255,353]. These algorithms approximate the energy or the time evolution of an eigenstate by performing measurements on a quantum state prepared according to a parameterized ansatz. A quantum autoencoder could be trained with states of size n_o qubits, obtained from a given ansatz, and later be used as a state preparation tool as described above. Because the autoencoder parameters are fixed after training, the only active parameters in the variational algorithm would be those associated to the preparation of a state in the latent space (n_l). Since $n_l < n_o$, the state preparation with autoencoders would require fewer parameters than the original ansatz.

In our specification of the autoencoder, we define the input states to be an ensemble of pure states, and the evolution of those states to be unitary. The most generalized picture of the autoencoder, however, would allow for inputs to be ensembles of mixed states and the set $\{U^{\vec{p}}\}$ to be a set of quantum channels. In the case of mixed state inputs, we remark that this formulation can in principle be captured by our model already. More specifically, one can consider the case where a set of ancillas (denoted A') representing a purification of the mixed state is input into the autoencoder along with the original input. Ulhmann's theorem[361] guarantees that there exists a purification whose fidelity is identical to that of the mixed state generated from tracing out the purification; namely, it is a maximum over a unitary V acting on the purification alone (although finding this unitary can be a difficult computational problem itself). Consider then the encoding $U^{\vec{p}}_{AB} \otimes V_{A'}$, where the original latent space is expanded to contain all of A' (i.e. none of the ancilla qubits are traced out). This purified system will recover the behavior of the mixed system. The autoencoder structure as defined here cannot completely capture the structure for general quantum channels, though we expect other computational tasks may be solved by considering specific

channel instances.

Are there any obvious limitations to the quantum autoencoder model? One consideration is that the von-Neumann entropy[361] of the density operator representing the ensemble $\{p_i, |\psi_i\rangle_{AB}\}$ limits the number of qubits to which it can be noiselessly compressed. However, finding the entropy of this density operator is not trivial – in fact, given a circuit that constructs a density operator ρ, it is known that, in general, even estimating the entropy of ρ is QSZK-complete[30]. This then opens the possibility for quantum autoencoders to efficiently give *some* estimate of the entropy of a density operator. ÂăIn a similar vein, the unitary of the autoencoder could be defined to include the action of a quantum channel, and the autoencoder used to give both an encoding for and some lower bound for the capacity of a quantum communication channel (although the trash state may no longer be useful for training the autoencoder in some of these cases).

It is natural to consider whether the quantum autoencoder structure we have defined is actually a generalization of its classical counterpart, as in the construction of[350]. It may certainly be possible that some particular family of unitaries, together with certain choices for n and k, can be constructed so that a mapping exists. However, it is unclear that such a correspondence would even be desirable. Rather, we believe the value of autoencoders in general lies in the relatively simple structure of forcing a network to preserve information in a smaller subspace, as we have defined here.

Another topic of interest for any quantum computing model is the computational complexity exhibited by the device. For our construction, it is clear that any complexity result would be dependent upon the family of unitaries that is chosen for the learning to be optimized over. As the training itself is based on classical optimization algorithms (with no clear 'optimal' learning method), this further obfuscates general statements regarding the complexity of the model.

5.6 APPENDIX

5.6.1 FIDELITY DERIVATION

Here, we show explicitly that the expression in Eq. (5.1) is the same as the fidelity term in Eq. (5.3). We may neglect the index i and $\vec{p}$, since the proof holds for all i and $\vec{p}$ individually. First, let us expand the fidelity in Eq. (5.1), recalling that $F(|\psi\rangle, \rho) = \langle\psi|\rho|\psi\rangle$ when $|\psi\rangle$ is a pure state and ρ is mixed[361],

$$\begin{aligned} F(|\psi\rangle, \rho^{out}) &= \langle\psi|\rho^{out}|\psi\rangle \\ &= \langle\psi| U_{AB'}^{\dagger} \operatorname{Tr}_B \left[U_{AB}(|\psi\rangle\langle\psi|_{AB} \otimes |a\rangle\langle a|_{B'}) U_{AB}^{\dagger} \right] U_{AB'} |\psi\rangle \\ &= \langle\psi'| \operatorname{Tr}_B \left[(|\psi'\rangle\langle\psi'|_{AB} \otimes |a\rangle\langle a|_{B'}) \right] |\psi'\rangle, \end{aligned} \tag{5.11}$$

again using the earlier notation $U_{AB}|\psi\rangle_{AB} = |\psi'\rangle_{AB}$. Now, expanding Eq. (5.3),

$$\begin{aligned} F(|\psi'\rangle_{AB}, \rho'_A \otimes |a\rangle_B) &= F(|\psi'\rangle, \operatorname{Tr}_{B'} \left[|\psi'\rangle\langle\psi'|_{AB'} \right] \otimes |a\rangle_B) \\ &= \langle\psi'| \operatorname{Tr}_{B'} \left[(|\psi'\rangle\langle\psi'|_{AB'} \otimes |a\rangle\langle a|_B) \right] |\psi'\rangle. \end{aligned} \tag{5.12}$$

Clearly, Eq. 5.11 and Eq. 5.12 are identical with the exception that the subsystem names of B and B' are reversed, which does not affect the equality.

6

A variational algorithm for device-tailored quantum error correction

Apart from minor modifications, this chapter originally appeared as:

"QVECTOR: an algorithm for device-tailored quantum error correction". Peter D Johnson, Jonathan Romero, Jonathan Olson, Yudong Cao and Alán Aspuru-Guzik, arXiv:1711.02249 (2017). Reproduced with permission from the authors.

6.1 Introduction

Uncontrollable environmental errors have remained the primary roadblock on the route to useful quantum information processing. Nevertheless, there is hope for achieving fault-tolerant quantum computation by implementing quantum error correction[58,87,122,181]. Fault-tolerant threshold theorems[6] show that, for a given degree of environmental noise, if each elementary operation can perform below a certain error rate, then concatenated quantum error correction schemes will out-pace error accumulation, enabling quantum computation to an arbitrary degree of accuracy.

The leading approaches to quantum error correction make use of topological stabilizer codes[45,87]. A major advantage of this approach is that the measurements used to diagnose errors may be performed on just a few neighboring qubits. Leading candidates for topological quantum error correction are the surface code[87,112], color code[45], and gauge color code[44]. Progress towards implementing surface and color codes experimentally has been demonstrated in[23,147] and[245],

respectively. Recent simulations have shown that the gauge color code[50] is expected to exhibit performance comparable to the previous schemes, though with the added benefit of transversal implementation of a universal gate set.

Unfortunately, these codes are not likely to be practical in near-term devices. Current estimates predict that the surface code will require 10^3 to 10^4 physical qubits to protect a single logical qubit[112]. In order to perform useful, fault tolerant quantum computation with near term devices, it seems that this number of physical qubits needs to be drastically reduced. We propose a path towards reducing such error-correction overhead via the design of device-tailored quantum error correcting codes.

In an actual device, quantum information is subject to hardware-specific quantum noise processes. Stabilizer codes are not optimal, in general, because they are not tailored to the specific noise of a given device[200]. Significant work in quantum error correction has investigated codes outside of the stabilizer formalism, which are tailored to noise beyond the Pauli error model[57,66,68,200,273]. Various schemes have been developed to numerically optimize encoding and decoding procedures with respect to a fixed noise model[109,110,184,185,329].

However, these optimization schemes are not applicable to useful quantum processing devices. First, they require a specific noise model as input to the optimization. Significant effort *has* been devoted to characterizing the noise of near-term devices[54,182,256] and quantum error correcting codes have been implemented as a tool for such characterization[191]. But, as larger systems are considered, the accuracy of these noise models is expected to drop rapidly[172].

Second, even if a sufficiently accurate noise model were known, existing classical processors are unable to handle the storage needed to carry out such optimization for near-term devices with 50 qubits[43]. The task of performing such an optimization seems better suited for a quantum processor.

Finally, even in cases where these optimizations can be performed, the optimized encoding and

recovery processes must be decomposed into a sequence of gates that are available on the device. It may be favorable, rather, to employ an optimization strategy which naturally integrates the constraints of the device's native gate set.

Recently, several hybrid quantum-classical (HQC) algorithms for solving specific optimization tasks have been developed. Two representative variational HQC algorithms are the variational quantum eigensolver (VQE)[255] and the quantum approximate optimization algorithm (QAOA)[104]. The former of these algorithms has been implemented experimentally on several quantum computing architectures[168,248,255,304]. Additionally, much theoretical work has been done to develop this genre of quantum algorithms[131,224,226,279,352,355]. A major appeal to such algorithms is that they operate successfully without the need for active quantum error correction and even show signs of suppressing certain types of errors[71,224,226].

Variational HQC algorithms are implemented by preparing quantum states as the output of a parameterized quantum circuit $U(\vec{p})$. Various ansatz states are repeatedly prepared and measured to collect outcome samples. The measurement data is classically processed and used to update the circuit parameters so as to optimize a particular cost function. As in the quantum autoencoder algorithm[280], the variational optimization of a circuit constitutes a quantum analogue of training a neural network in machine learning.

We propose a variational HQC algorithm for designing device-tailored error corrected quantum memories. This approach naturally points to an extension for designing error corrected gates. The algorithm solves a number of problems which hamper the classical optimization schemes for the same task. First and foremost, our proposal forgoes the need for a noise model because the optimization is carried out *in situ* and the noise perfectly simulates itself on the device. Second, the optimization step is not necessarily hindered by the exponential scaling of the Hilbert space dimension in the same way that the previous proposals are. Measurement statistics, obtained using

the device as a quantum sampler, are used to approximate the average fidelity of the encoding-decoding scheme. This average fidelity serves as the cost function for the classical optimization algorithm. Finally, by constructing the variational circuits out of a device-native gate set, the optimized encoding and recovery processes are manifestly decomposed into an implementable sequence of gates.

6.2 Background: quantum error correction

The Pauli group stabilizer formalism of quantum error correction [122] has earned its place as the most popular approach to quantum error correction. By describing all mathematical objects in terms of elementary gates, the formalism has enabled significant theoretical analysis and experimental implementation. Our proposal, however, is not based on the Pauli group stabilizer formalism. Accordingly, we review a more general framework of quantum error correction.

The general mathematical formalism of subspace code quantum error correction [180] is summarized as follows. First, k qubits of logical quantum information $\mathcal{H}_L \simeq \mathcal{Q}^{\otimes k}$ are encoded via an encoding process $\mathcal{E}$ into n physical qubits $\mathcal{H}_P \simeq \mathcal{Q}^{\otimes n}$. Next, the physical qubits are subjected to some noise process $\mathcal{N}$. Finally, the quantum information is attempted to be recovered by a decoding process $\mathcal{D}$. The quality of the quantum error correction scheme can be characterized by how well the sequence of processes approximates the identity channel $\mathcal{D} \circ \mathcal{N} \circ \mathcal{E} \approx \mathcal{I}$, which may be quantified by either the average fidelity or worst-case fidelity of the quantum process, or by some other figure of merit.

It is standard to use an encoding in which encoded states are pure: $\mathcal{E}(|\psi\rangle\langle\psi|) = |\overline{\psi}\rangle\langle\overline{\psi}|$. The linear span of state vectors in the range of $\mathcal{E}$ is referred to as the code space $\mathcal{C}$. The code space is a 2^k-dimensional subspace of the physical Hilbert space, $\mathcal{C} \leq \mathcal{H}_P$. It is instructive to factor the code

space into a logical subsystem $\mathcal{H}_{\bar{L}}$ and a syndrome subsystem $\mathcal{H}_S$,

$$\mathcal{C} \simeq \mathcal{H}_{\bar{L}} \otimes \text{span}(|s_0\rangle) \leq \mathcal{H}_P \simeq \mathcal{H}_{\bar{L}} \otimes \mathcal{H}_S, \tag{6.1}$$

where $|s_0\rangle$ is a fixed state of $\mathcal{H}_S$. This factorization does not correspond to a separation of physical qubits, but rather to a separation of *virtual subsystems*[264,371], and is the key structure of the *subsystem principle* of quantum error correction[180].

The encoding should be chosen to match the features of the noise model. It is standard to model the noise as a completely positive trace-preserving map, expressed in Kraus representation as $\mathcal{N}(\cdot) = \sum_j K_j \cdot K_j^\dagger$. Perfect recovery is possible if and only if there exists an encoding $\mathcal{C}$ such that each Kraus operator satisfies

$$K_j |\overline{\psi}\rangle = V^\dagger (|\psi\rangle \otimes |\tau_j\rangle) \tag{6.2}$$

for all $|\overline{\psi}\rangle \in \mathcal{C}$ and for a fixed unitary $V^\dagger$ and some unnormalized $|\tau_j\rangle$ which depend on K_j; note that this is simply a rephrasing of the Knill-Laflamme condition[181] for exact correctability. Conditional upon the syndrome system $\mathcal{H}_S$ being initialized in $|s_0\rangle$, the logical quantum information is protected from the noise in the virtual subsystem $\mathcal{H}_{\bar{L}}$. If errors are to be corrected while the quantum information is still encoded, $V^\dagger$ is inverted by the application of V and the syndrome system is reset back to $|s_0\rangle$. To decode, the logical subsystem is mapped back into the k-qubit system $\mathcal{H}_L$, and the syndrome qubits are simply traced out.

To briefly make contact with the stabilizer formalism, the code is defined by the intersection of eigenvalue-1 subspaces of the Pauli stabilizer operators $\{S_j\}$, which admit the decomposition $S_j = \mathbb{I}_{\bar{L}} \otimes (\tilde{S}_j)_S$, while the logical operators $\{\overline{Z}_i, \overline{X}_i\}$ of the code decompose as $\overline{Z}_i = (Z_i)_{\bar{L}} \otimes \mathbb{I}_S$. For a more thorough account of this connection, see[264] or[180].

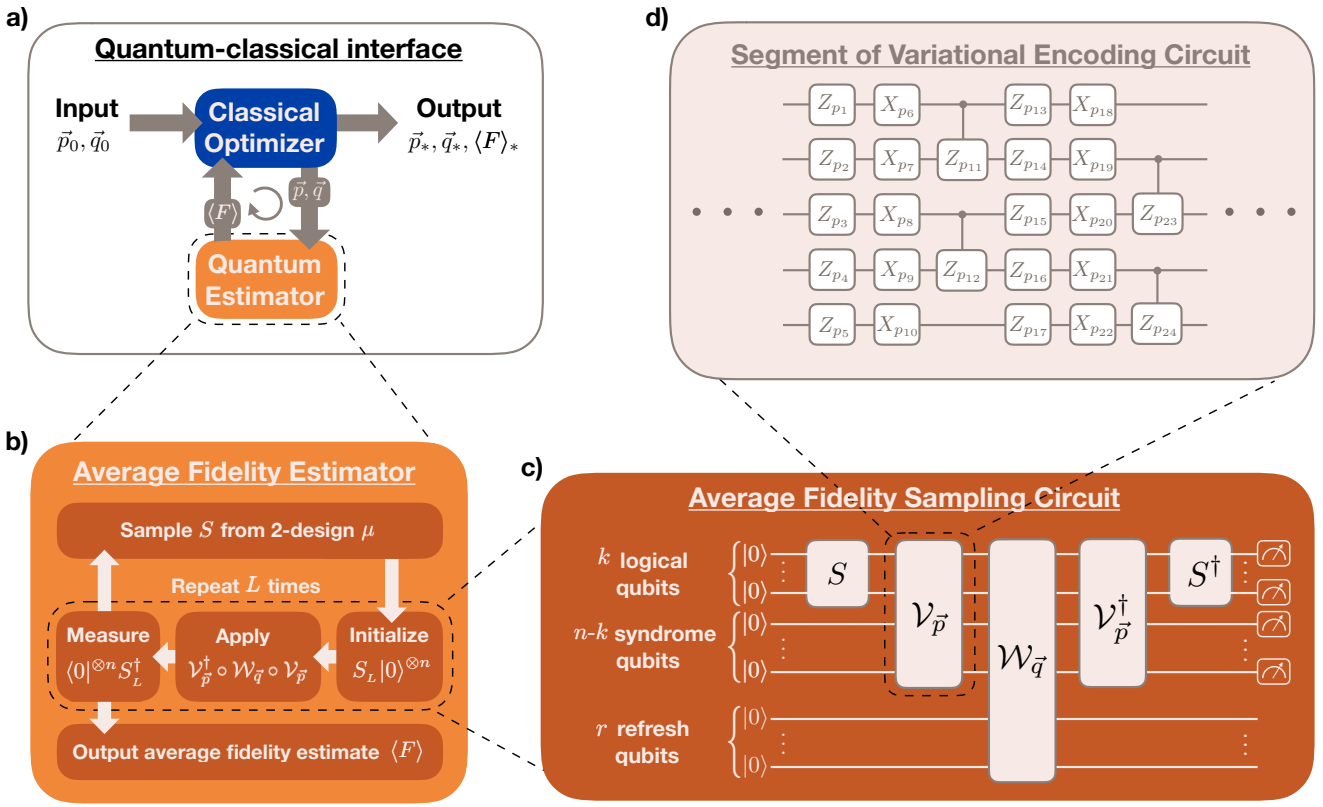


Figure 6.1: Training stage schematic of the QVECTOR algorithm. a) QVECTOR uses a classical optimizer to optimize a function whose output value is determined by calling a quantum subroutine, the quantum average fidelity estimator. b) The quantum average fidelity estimator uses a variational quantum circuit to send random-sampled states $S|0\rangle^{\otimes k}$ through the circuit $\mathcal{W}_{\vec{q}} \circ \mathcal{V}_{\vec{p}}$ and records the measured channel fidelity of each state as 0 or 1. The average of these measured channel fidelities is output and fed back to the classical optimizer. c) Within each call to the average fidelity estimator, the quantum circuit is run L times. In each run, the state $|0\rangle^{\otimes n}$ is initialized on k logical qubits and $n-k$ syndrome qubits, the k logical qubits are transformed by the 2-design sampled unitary S, the noisy encoding-decoding circuits $\mathcal{V}_{\vec{p}}$ and $\mathcal{W}_{\vec{q}}$ are applied, the inverse of S is applied, and the k logical qubits are measured in the computational basis. d) The variational circuit $\mathcal{V}_{\vec{p}}$ and $\mathcal{W}_{\vec{q}}$ may be parameterized as seen fit by the particular device. One example of $\mathcal{V}_{\vec{p}}$, constructed from single-qubit X- and Z-rotations, and 2-qubit controlled-Z rotations is depicted; here, the variational parameters, $p_1, p_2, \ldots$, are the rotation angles of each circuit element.

6.3 Quantum variational error correction algorithm (QVECTOR)

The objective of the quantum variational error corrector (QVECTOR) algorithm is to output an encoding and recovery circuit which sufficiently correct errors in a quantum memory. First we describe the protocol used for optimizing, or *training*, the encoding-recovery circuits, then we will describe their implementation as an error correction scheme.

The physical qubits are divided into two sets, qubits which will encode k logical qubits and r qubits which facilitate the non-unitary error-recovery process. The encoding and recovery are implemented with a circuit of tunable gates. Before encoding, the first k qubits are prepared in a to-be-encoded state $|\psi\rangle$ and the remaining $n-k$ qubits are initialized in a fiducial state $|0\rangle^{\otimes n-k}$. The sequence of gates $V(\vec{p})$ then acts to encode the state of the first k qubits into n qubits. The recovery process is aided by an additional register of r "refresh" qubits, and is implemented by a sequence of gates $W(\vec{q})$ coupling all $n+r$ qubits.

The figure of merit, or cost function, we use to evaluate an encoding-recovery pair is the *average code fidelity*, based on the quantum channel Haar average fidelity [82]. For a (possibly) noisy recovery operation $\mathcal{R}$, the average code fidelity is defined as

$$\langle F \rangle \equiv \int_{\psi \in \mathcal{C}} \langle \psi | \mathcal{R}\left(|\psi\rangle \langle \psi|\right) |\psi\rangle \, d\psi, \tag{6.3}$$

where the integral is performed with respect to the Haar distribution of states in the code space. In addition to evaluating the preservation of the logical qubits, it scores the ability of the encoded recovery circuit $W(\vec{q})$ to properly return the quantum information to the code space. A well-performing encoded recovery operation can be applied in sequence to extend the lifetime of the quantum memory.

As a hybrid quantum-classical algorithm, QVECTOR uses a quantum and a classical processing

unit working in tandem. The objective of the classical processing unit is to *optimize* the average code fidelity with respect to the tunable circuit parameters $(\vec{p},\vec{q})$, while the quantum information processing unit is called by the classical processor as a subroutine to *estimate* the average code fidelity associated with the given encoding-recovery pair $(V(\vec{p}),W(\vec{q}))$.

The average code fidelity estimation procedure we use is inspired by the the sampling approach of[82,97]. Assuming S and $S^\dagger$ are efficient to implement, the input-output fidelity of any term in Eq. (6.3) can be efficiently computed by preparing $|0\rangle^{\otimes k}|0\rangle^{\otimes n-k}$, applying state preparation S on the first k qubits, performing the encoding-decoding operation $(V(\vec{p})_n^\dagger\otimes\mathbb{I}_r)W(\vec{q})(V(\vec{p})_n\otimes\mathbb{I}_r)$, applying the inverse of S, and then measuring the first n qubits in the computational basis. After N trials, the fraction of all-0 outcomes estimates the fidelity of $\mathcal{R}(|\psi\rangle\langle\psi|)$ with respect to $|\psi\rangle\langle\psi|$ with standard deviation $\mathcal{O}(\frac{1}{\sqrt{N}})$.

To obtain an estimate of the *average* code fidelity, we could vary the state preparation circuit S, and obtain code fidelity data for sufficiently many samples S drawn from the Haar distribution. However, because Haar-random sampling is not efficient[243] and because the average code fidelity depends only on the second moment of the distribution, we instead sample S from an efficiently implementable unitary 2-design[82]. A unitary 2-design is a measure μ on the unitary group $\mathcal{U}(d)$ satisfying

$$\int_{\mathcal{U}(d)} S^{\otimes 2}\otimes S^{\dagger\otimes 2}\,d\mu(S)=\int_{\mathcal{U}(d)} U^{\otimes 2}\otimes U^{\dagger\otimes 2}\,dU. \tag{6.4}$$

With a 2-design μ, the average code fidelity of the encoding-decoding is written as

$$\langle F\rangle\equiv\int_{\mathcal{U}(d)}\langle 0^{(n)}|S^\dagger\mathcal{V}_{\vec{p}}^\dagger\mathcal{W}_{\vec{q}}\mathcal{V}_{\vec{p}}\left(S|0^{(n)}\rangle\langle 0^{(n)}|S^\dagger\right)S|0^{(n)}\rangle\,d\mu(S), \tag{6.5}$$

where $\mathcal{V}_{\vec{p}}$ and $\mathcal{W}_{\vec{q}}$ denote the physically implemented quantum channels—noisy versions of the parameterized circuits. This quantity may be estimated with standard deviation N after $\mathcal{O}(\frac{1}{\sqrt{N}})$

trials as follows. In each trial, S is sampled from the 2-design and implemented in the process $\mathcal{S}^\dagger \mathcal{V}_{\vec{p}}^\dagger \mathcal{W}_{\vec{q}} \mathcal{V}_{\vec{p}} \mathcal{S}$ that is applied to the initial state $|0\rangle^{\otimes n}$. In each trial, the first n qubits are measured in the computational basis and the number of all-0 outcomes over N constitutes an unbiased estimator for the average code fidelity.

In some cases, it may be favorable to implement an approximate unitary 2-design. A good candidate is the recent construction of an ε-approximate 2-design[239], which is particularly simple to implement. Using this approximate 2-design, each quantum measurement is interpreted as returning a binary sample from a *biased estimator* of the true average fidelity. As shown in Appendix 6.6.3, using ℓ applications of the randomization circuit in[239], the bias of this estimator is upper bounded by $\frac{2^{k(\ell+1)}+2^{k\ell}-2}{2^{2k\ell}(2^k-1)} \sim \mathcal{O}(1/2^{k\ell})$. Thus, after N samples from this biased estimator, the estimated average fidelity is expected to deviate from the true average fidelity as $\mathcal{O}(\frac{1}{\sqrt{N}} + \frac{1}{2^{k\ell}})$.

A schematic of this fidelity estimation algorithm is depicted in Figure 6.1a. The quantum fidelity estimation algorithm serves as a cost function evaluation subroutine which is called by the classical processor that performs an optimization such as LBFGS[246] or SPSA[320]. Any "quantum speedup" realized by this algorithm is due to the efficiency with which a quantum circuit can be used to estimate its own average fidelity*.

After the circuit is trained to a sufficient average fidelity, it is ready to be used as a quantum error correction scheme for preserving a quantum memory. Once the quantum information is encoded using $V(\vec{p})$, the recovery circuit $W(\vec{q})$ may be applied at regular time intervals to recover from errors accrued in the memory. Before each recovery step, the refresh qubits must be reset to the fiducial state $|0\rangle^{\otimes r}$.

*Note that the analysis in this chapter does not unequivocally prove that such a speedup is possible with this approach. As this method is not easily amenable to theoretical analysis, a proper evaluation of its effectiveness will come from physical implementation.

6.4 NUMERICAL SIMULATIONS

Towards evaluating the effectiveness of this algorithm, we simulate its performance on several few-qubit examples in the presence of simple noise models. Against three qubit phase-damping error, we find that the algorithm is able to learn an encoding and recovery map which perform nearly as well as the optimal phase-error code and recovery process. Considering a more realistic noise process by incorporating amplitude damping[116] on five qubits, we find that our algorithm can learn useful error correction which exploits coherence in the Pauli errors where the five-qubit code fails to improve the physical-qubit fidelity.

All noise in the system is modeled as a quantum channel which acts after the encoding map and before the recovery process. As such, the state preparation, parameterized circuit gates, and measurements are taken to be ideal. For a noise channel $\mathcal{N}$, the optimization cost function is defined as the average code fidelity of the quantum process $\mathcal{V}_{\vec{p}}^{\dagger}\mathcal{W}_{\vec{q}}\mathcal{N}\mathcal{V}_{\vec{p}}$.

The variational circuits consist of layers of single qubit rotations interleaved with two-qubit entangling operations. An example is shown in Figure 6.1.d., where the variational circuits consist of single-qubit Pauli-X and Pauli-Z rotations as well as nearest-neighbor controlled-Z rotations, whereby the variational parameters $\vec{p}$ and $\vec{q}$ control the rotation angles of each of these gates. The specific form of the variational circuits we use is described in Appendix 6.6.2. As the QVECTOR algorithm is agnostic with respect to the choice of variational circuit structure, we chose to simulate circuits which can be implemented natively in existing hardware[23]. The classical optimization can be performed using a variety of methods, including SPSA[320], basin-hopping[347], or Bayesian optimization[232]. However, the reported data was obtained using the quasi-Newton method L-BFGS[246].

Three-qubit phase-damping- In the first simulation, our goal is to analyze the performance of a quantum memory with active recovery learned by the QVECTOR algorithm. We consider a setting

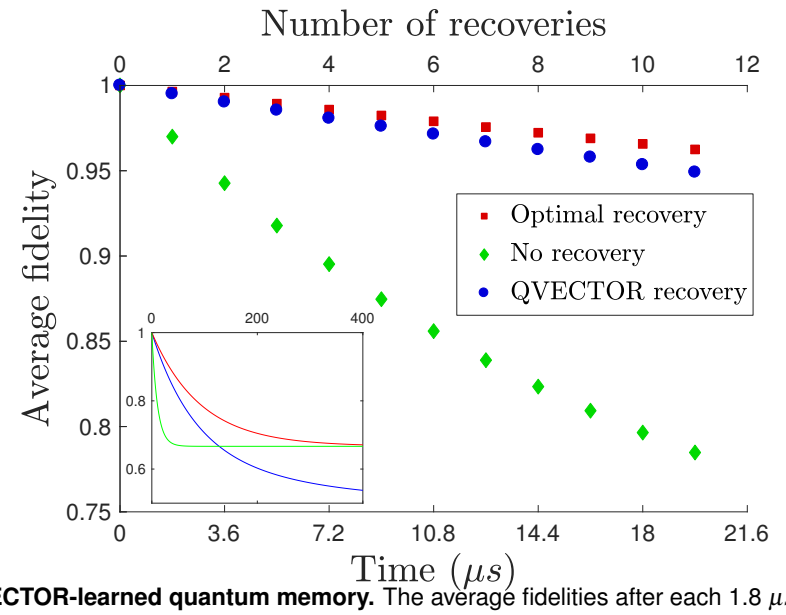


Figure 6.2: QVECTOR-learned quantum memory. The average fidelities after each 1.8 μs recovery step are plotted for various procedures, showing that the QVECTOR-learned recovery nearly matches that of the optimal recovery procedure, as given by the standard phase damping code. In the no-recovery case (i.e. decoherence of a single physical qubit), the noise process in each step is modeled by a probabilistic phase damping with error probability $p = 0.045$, corresponding to $T_2 = 19\ \mu s$[23]. To account for additional error due to noisy gates in the optimal and QVECTOR recoveries, the noise here is modeled by a probabilistic phase damping with error probability $p = 0.091$. Despite the addition of gate error, the QVECTOR recovery extends the effective T_2 by nearly six-fold to $\sim 110\ \mu s$, while the optimal recovery extends this to $\sim 165\ \mu s$. The QVECTOR recovery circuit uses just 30 layers of two-qubit gates, which is comparable to the number used in the optimal recovery circuit. The inset depicts the many-recovery limit, where the QVECTOR average fidelity eventually drops below the no-recovery average fidelity after roughly 150 recovery steps, possibly due to systematic over-rotation in the learned recovery process.

where a single qubit is encoded into three qubits which are subject to independent probabilistic phase damping. Two additional (noise-free) qubits are used to facilitate an encoded recovery operation. As described in Appendix 6.6.1, the phase-flip error rate is chosen to match the specifications of a sequence of realistic one- and two-qubit gates implemented with Xmon qubits[23] corresponding to the particular parameterized circuit we employ (see Appendix 6.6.2). We find that this corresponds to a single qubit phase-flip probability of $p = 0.091$ and requires a duration of $1.8\ \mu s$. There are two points of reference for assessing the performance of QVECTOR. The first is the case of no error correction, where a single physical qubit is used as a quantum memory. As outlined in Appendix 6.6.1, to account for the lack of noisy gates in this case, the error rate in the no-encoding case is decreased to $p = 0.045$ per time step. The second point of reference is the case

of optimal encoding and recovery with respect to the $p = 0.091$ phase-damping process. The standard phase-error code and corresponding recovery optimize the average fidelity metric with respect to a general phase-damping process.

As described in detail in Appendix 6.6.2, $V(\vec{p})$ and $W(\vec{q})$ are trained with respect to the above noise model. The encoding and recovery pair we use was selected as the optimally performing scheme among twelve distinct training attempts. To simulate the performance of these optimized circuits as a quantum memory, we compute the average fidelity of the process $\mathcal{V}_{\vec{p}}^{\dagger}(\mathcal{W}_{\vec{q}}\mathcal{N})^{M}\mathcal{V}_{\vec{p}}$ for a various number of iterations M. As shown in Fig. 6.2, we find that, with respect to phase-damping noise, the simulated QVECTOR algorithm results in a quantum memory that has an effective T_2 time of approximately 110 μs, nearly six times that of the bare physical qubit. This shows that, although the gates used to implement the QVECTOR recovery circuit are modeled as to incur additional noise, there is, nonetheless, an improvement in the lifetime of the quantum memory.

For the first ten or so recovery steps, the QVECTOR average fidelity remains comparable to that of the optimal recovery. However, as shown in the inset of Figure 6.2, the QVECTOR average fidelity equilibrates to $\langle F \rangle = 1/2$ as opposed to $\langle F \rangle = 2/3$, dropping below the no-encoding curve after roughly 150 iterations. We conjecture that this is due to a systematic over-rotation in the recovery process which accrues over many repeated recoveries. We obtained evidence for this explanation by examining the zero-noise limit and finding that the average fidelity in this case undergoes damped harmonic oscillation with a period of 1089 recovery steps. By training on just a single recovery step, such an over-rotation is indistinguishable from incoherent error. This points to the possibility of mitigating such over-rotation error by training on multiple recoveries, as discussed in the outlook section.

Five-qubit communication setting with amplitude- plus phase-damping error- The goal of the second simulation is to test the simulated QVECTOR performance against a more realistic noise

model and in a different error-correction setting. In some instances of a quantum memory, such as during transmission of a quantum state during communication, active error recovery is impractical or unavailable. This situation arises, for instance, if one were attempting to relay qubits through an optical fiber or transport qubits between two neighboring quantum processors. If one cannot repeatedly apply a recovery channel during transmission, the best error-reduction one can hope to achieve is an optimized encoding at the source followed by a single decoding at the destination. We investigate the performance of the QVECTOR algorithm in such a case, determining the average fidelity for various "wait times" corresponding to the delay between transmission and reception when the state is subject to error.

In the single decoding scenario, the quantum information does not need to be returned to the code by the recovery step. Rather, the encoded quantum information only needs to be decoded back to the first physical qubit. Thus, a unitary correction suffices, and the refresh qubits are unnecessary (i.e. $r = 0$). In this setting, we analyzed QVECTOR's performance where $k = 1$, $n = 5$, and $r = 0$ subject to independent continuous amplitude- plus phase-damping (APD) for various wait times with $T_1 = 57\ \mu s$ and $T_2 = 19\ \mu s$, as described in Appendix 6.6.1.

As shown in Fig. 6.3, the simulated QVECTOR algorithm learns an encoding and recovery pair with average fidelity greater than a physical qubit subject to the same noise process. We also compare to the standard five qubit code, which is known to be optimal for depolarizing noise[273]. We find that, although the five qubit code fails to be useful after $t = 3.5\ \mu s$, by training QVECTOR for a 4 μs wait time, the encoding-decoding circuit learned by QVECTOR outperforms the no-encoding average fidelity for all wait times considered. The numerically-optimized average fidelity is obtained using the iterated semi-definite programming method of[184] (see Appendix 6.6.2), and plotted for comparison. Through training, the encoding-decoding pair was selected as the optimally performing scheme among three distinct training attempts.

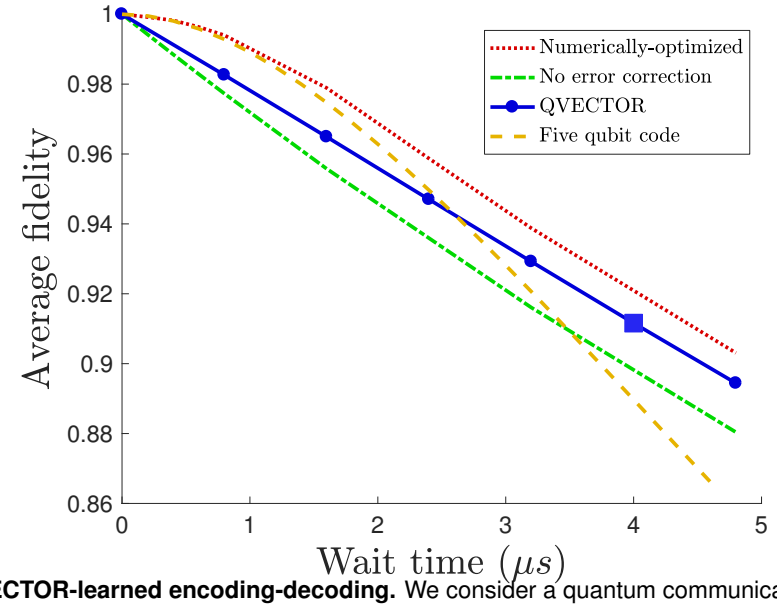


Figure 6.3: QVECTOR-learned encoding-decoding. We consider a quantum communication setting, where only a single recovery-decoding step is available after a pre-determined "wait time". For each wait time, the average fidelities of various encoding-decoding procedures are plotted, showing that the QVECTOR-learned encoding-decoding scheme continues to be useful beyond the time at which the standard five-qubit code fails to be so. Between encoding and decoding, the noise is modeled as a continuous amplitude- plus phase-damping channel with $T_2 = 19\ \mu s$ and $T_1 = 57\ \mu s$ (see Appendix 6.6.1). The numerically-optimized encoding-decoding pairs are obtained using the iterated semi-definite programming method of [184]. The QVECTOR-learned encoding-decoding pair were initially trained for the $4\ \mu s$ wait time. This QVECTOR-optimized encoding-decoding pair for $4\ \mu s$ was used as an initial point for gradient-based optimization of the remaining wait times. The QVECTOR encoding-decoding pairs continue to be useful beyond the point at $3.5\ \mu s$ where the five-qubit code drops below the no-encoding average fidelity.

Finally, we investigated the potential of the QVECTOR algorithm for discovering encoding-decoding circuits which exploit structure in the noise in ways that stabilizer codes do not. In the amplitude- plus phase-damping model, the Kraus operators are coherent superpositions of Pauli operators. A common technique in simulating noise is to ignore such coherences, and represent the process as an incoherent mixture of Pauli errors. This simplification is referred to as the Pauli-twirling approximation (PTA) [116].

There is mounting evidence in the literature that, in order to compute the performance of stabilizer codes under realistic noise processes, it suffices to compute their performance against a PTA version of the corresponding noise channel [132,335]. However, this condition holds only for small error rates [173]. At higher error rates and in the presence of coherent errors, error correction

Exploiting coherence between Pauli errors

Scheme	Exact APD	PTA-APD
QVECTOR*	0.912	0.832
Numerically optimized	0.920	0.904
No-encoding	0.898	
Five qubit code	0.890	

* The same encoding-decoding is used for both Exact APD and PTA-APD.

Table 6.1: This table compares the average fidelities achieved by various error correction schemes for the amplitude- plus phase-damping noise model at wait time 4 μs. The right-most column reports the performance of the scheme against an approximate noise model, which ignores coherence among Pauli errors, obtained by Pauli-twirling[116] the noise channel (see Table 6.2). While the five qubit code (red) clearly does not take advantage of the coherence in the amplitude- plus phase-damping (APD) channel, the encoding-decoding obtained by QVECTOR can be shown to genuinely make use of this coherence. The average fidelity achieved by the QVECTOR encoding-decoding (green) is, not surprisingly, diminished when computed against a Pauli-twirling approximation (PTA) of the APD noise. To show that the QVECTOR encoding-decoding unequivocally exploits the Pauli error coherence, we compare to the average fidelity of the numerically-optimized encoding-decoding, subject to the PTA-APD noise model, where Pauli-error coherence is removed (orange). The discrepancy between these two shows that the QVECTOR encoding-decoding exploits the coherence between Pauli errors in the noise model to achieve an average fidelity that could not be reached without such coherence.

schemes constructed around the Pauli-error model can lead to fidelities even worse than those obtained without encoding (c.f. Fig. 6.3).

We considered the APD noise model acting for 4 μs with $T_1 = 57\ \mu s$ and $T_2 = 19\ \mu s$, matching the parameters used in Fig. 6.3. As found in Table 6.2, the Kraus operators of the APD channel are coherent superpositions of Pauli errors, while the Pauli-twirled approximation of the APD channel (PTA-APD) constitutes an incoherent mixture of Pauli errors. An important physical difference between these two channels is that the former is non-unital, enabling T_1 decay to the ground state.

As shown in Table 6.1, we computed the average fidelity for various encoding-decoding schemes subject to APD and to PTA-APD. For the QVECTOR case we used the encoding-decoding obtained for the 4 μs APD noise model as reported in Fig. 6.3. The performance of this scheme (green) relies significantly on the coherence between the Pauli errors in the APD noise model, as evidenced by the discrepancy between its APD and PTA-APD average fidelities. In contrast, the five qubit code does not utilize the coherence among the Pauli errors

(red). It remains to verify that the average fidelity achieved by QVECTOR cannot be obtained by any encoding-decoding if the Pauli-error coherence is removed from the noise model. We performed the bi-convex optimization method of[184] to numerically determine the maximal average fidelity that can be achieved among all encoding-decoding schemes subject to PTA-APD (orange). Comparing this to the QVECTOR value, we find that the QVECTOR encoding-decoding exploits the coherence between Pauli errors in that this average fidelity cannot be achieved by any encoding-decoding scheme if Pauli-error coherence were removed from the noise process.

This finding highlights the fact that codes designed to be agnostic to coherence among Pauli errors, such as many stabilizer codes, fail to exploit such structures. In contrast, the QVECTOR methodology does not build in this limitation, as it does not assume a noise model *a priori*. Therefore, this approach may be able to outperform other approaches by exploiting structure, such as Pauli-error coherence, in the noise processes.

6.5 Conclusions and outlook

We developed a hybrid quantum-classical algorithm which learns encoding and error-recovery processes tailored to the noise of the target quantum device. The opportunities for improvement over leading quantum error correction techniques are three-fold. First, by using a native parameterized gate set, this approach may facilitate a more-effective use of available resources for realizing quantum error correction. Second, compared to previous optimization-based approaches, the optimization algorithm in QVECTOR is, in principle, scalable: the simulation of the noise process is efficient and accurate, as the device perfectly simulates its own noise, while the average code fidelity estimation for assessing performance is efficient by using randomized benchmarking-like techniques. Finally, unlike other approaches to error correction, QVECTOR does not assume any error model *a priori*. In contrast, it tailors the encoding and recovery

processes to the noise inherent in the device, which might allow to correct errors beyond the Pauli-error model used by stabilizer codes.

We simulated two distinct five-qubit examples of the QVECTOR algorithm which can be performed with existing hardware. In the first case, we simulated an encoding of a single qubit into three physical qubits subject to realistic rates of phase damping noise. Two ancillary qubits are used to implement the encoded recovery. The simulated QVECTOR algorithm learns an encoding and recovery process which extends the T_2 of the quantum memory from $\sim 19\ \mu s$ to $\sim 110\ \mu s$. Additionally, we considered a quantum communication setting, in which there is no active error correction, but, instead, the quantum information is recovered after a known wait time. Here, we simulated five qubits subject to a combination of continuous amplitude- plus phase-damping noise for various durations. We found that the QVECTOR-learned encoding-decoding pairs continue to bear an advantage beyond wait times of $3.5\ \mu s$, the point where the five qubit code fails to be useful. By testing the QVECTOR-learned encoding-decoding pairs under a Pauli-twirled approximation of the same error model, we found that QVECTOR may outperform standard stabilizer codes by exploiting coherence among Pauli errors.

Although promising, the simulation results neither prove the algorithm's scalability nor render it impractical. Reaching such conclusions will likely require an empirical approach. The outcome will be largely dependent on the nature of the cost function landscape (i.e. the estimated average code fidelity as a function of the circuit parameters) and the classical optimization methods that are used to explore it. It is possible that, as the system size is scaled up, the cost function landscape becomes increasingly proliferated with poor local optima or saddle points.

Conversely, and more optimistically, it is possible that realistic noise processes will possess more structure to exploit than the simulated noise models. This could result in a cost function landscape with many favorable local optima. An inspiring example is given in recent work where

the structure of correlated noise is exploited to design non-standard quantum error correction schemes for quantum sensing[195]. Crucially, the nature of this cost function landscape is highly dependent on the choice of variational circuit structure, highlighting the importance of designing effective variational circuits.

There are several issues which remain to be investigated. First, the algorithm should be simulated against a more realistic noise model which takes into account the error of each gate, including those of state preparation and measurement. There will be stochastic errors in the cost function estimation not yet accounted for due to finite sampling error and state preparation and measurement (SPAM) errors. For the latter, it may be favorable to use randomized benchmarking techniques[216], fitting a function of the fidelity estimation for various circuit iterations to obtain a more accurate error rate per iteration estimate, mitigating the offset due to SPAM errors.

There are several other metrics for scoring the performance of a quantum error correction scheme in addition to the simple-to-compute average fidelity. One figure of merit, known as the worst-case fidelity, or simply the error rate[286], is considered to be a more faithful metric for the quality of a quantum process[348], although much more difficult to calculate. However, recent work has shown that worst-case fidelity, along with average fidelity fail to properly assess the performance of error correcting schemes in some cases[362]. It remains to determine which metrics will ultimately be the most useful in practice.

The most important future direction is the augmentation of the existing QVECTOR algorithm to enable learning error-corrected quantum gates for fault-tolerant quantum computation. Towards this, we may view the current work as a step in this direction, as it provides evidence that simple error-corrected gates (i.e. the identity gate) may be variationally constructed. One could imagine training the encoded recovery circuit to be able to apply a target elementary logical gate, such as a two-qubit CNOT gate. In principle, a polynomially-sized universal gate set could be learned in this

manner, and then implemented for quantum computation. The efficacy of this approach will be determined by the performance of such gates under composition.

Concatenation of quantum error correcting codes provides the basis for the standard model of quantum computation. It remains to explore concatenated variational error correcting codes. As an example, one could imagine learning several five-qubit variational codes and corresponding recovery circuits, placing five of these in parallel, and applying another round of variational encoding on these twenty-five qubits.

Finally, the most encouraging feature of this approach to quantum error correction is that it can be implemented with existing hardware. Although it lacks the beauty of stabilizer QEC and does not boast any theoretical promises à la threshold theorems, QVECTOR is sufficiently different from standard approaches that it may provide a fresh avenue of exploration towards realizing quantum computation.

6.6 Appendix

6.6.1 Noise models

We simulated the action of noise channels using the standard Kraus operator formalism. In the one qubit case, the action of the channel in the density matrix is given by the operation

$$\mathcal{N}(\rho) = \sum_{j=1}^{m} K_j \rho K_j^\dagger \tag{6.6}$$

where ρ_i is the initial density matrix and K_j are the corresponding Kraus operators satisfying the completeness relation $\sum_{j=1}^{m} K_j^\dagger K_j = I$. To simulate the effect of noise on an n-qubit register, we assumed an independent noise model, where the Kraus operators correspond to a tensor products

of single qubit Kraus operators. The effect of noise on the n-qubit register can be written as:

$$\mathcal{N}^{\otimes n}(\rho) = \sum_{j_1,\cdots,j_N} \left(\bigotimes_{i=1}^{n} K_{j_i} \right) \rho \left(\bigotimes_{i=1}^{n} K_{j_i}^{\dagger} \right), \tag{6.7}$$

where the sum runs over all the possible m^N tuples of $j_1, \cdots, j_N$ indexes.

We consider two different noise channels in our QVECTOR simulations: phase-damping (PD) and a combination of amplitude damping and phase damping (APD). Both are captured by standard $T_{1,2}$ decoherence according to the following map:

$$\rho = \begin{bmatrix} 1-\rho_{11} & \rho_{01} \\ \rho_{01}^{*} & \rho_{11} \end{bmatrix} \rightarrow \tag{6.8}$$

$$\begin{bmatrix} 1-\rho_{11}e^{-t/T_1} & \rho_{01}e^{-t/2T_1}e^{-t/T_\phi} \\ \rho_{01}^{*}e^{-t/2T_1}e^{-(t/T_\phi)} & \rho_{11}e^{-t/T_1} \end{bmatrix} \tag{6.9}$$

where $\frac{1}{T_\phi} = \frac{1}{T_2} - \frac{1}{2T_1}$. The Kraus operators for each channel are described in Table 6.2. The parameters γ and λ are computed with respect to the experimental parameters corresponding to the wait time of the decoherence, (t_{step}), the decay time, (T_1), and the dephasing time, (T_2), according to the equations $\gamma = 1 - e^{-t_{\text{step}}/T_1}$ and $\lambda = e^{-t_{\text{step}}/T_1} + e^{-2t_{\text{step}}/T_2}$. We also employed the Pauli-twirled version of the APD channel, denoted as PTA-APD, as described in [116]. The PD channel is obtained by assuming $t_{\text{step}}, T_2 \ll T_1$ and the phase-error probability per time step is $p = (1 - e^{-t_{\text{step}}/T_2})/2$. This assumption does not fully capture the specifications of the Xmon qubits which we are basing the simulations on [22,23]. This model, therefore, is chosen as a simple starting point from which more-sophisticated explorations should be carried out.

In our simulations, we chose noise specifications to match error rates of the physical processes arising in state of the art superconducting qubit architectures. In the first simulation, we modeled

Noise channel	Kraus Operators
Phase damping (DP)	$K_1 = \sqrt{p}I; \quad K_2 = \sqrt{1-p}Z$
Amplitude- plus Phase-Damping (APD)	$K_1 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma-\lambda} \end{bmatrix}; K_2 = \begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix}; K_3 = \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{\lambda} \end{bmatrix}$
Pauli twirled - APD (PTA-APD)	$K_1 = (1-p_X-p_Y-p_Z)I; \quad K_2 = p_X X; \quad K_3 = p_Y Y; \quad K_4 = p_Z Z$

Table 6.2: Kraus Operators for common one-qubit error channels employed in classical simulations of the QVECTOR protocol. p is the error rate for the phase damping process. The parameters γ and λ are associated to the amplitude damping and phase damping processes, respectively, and are computed from the process time t_{step} and the T_1 and T_2 times as $\gamma = 1 - e^{-t_{\text{step}}/T_1}$ and $\lambda = e^{-t_{\text{step}}/T_1} + e^{-2t_{\text{step}}/T_2}$. The parameters for the Pauli-twirled approximation (PTA) of APD are calculated as $p_X = p_Y = \frac{\gamma}{4}$ and $p_Z = \frac{1}{2} - \frac{\gamma}{4} - \frac{\sqrt{1-\gamma-\lambda}}{2}$ [116].

the noise as independent single-qubit PD channels, acting on the three code-carrying qubits prior to the recovery circuit. The single-qubit phase-error probability per recovery step was set so as to incorporate the error incurred by each gate in the recovery circuit as follows.

The three-qubit noise process $\mathcal{N}$ was constructed from a circuit of single qubit phase-flip processes, which correspond to the unitary gates in the recovery circuit. To each single-qubit gate in the recovery circuit, we associated a corresponding single-qubit phase-flip noise process. The phase-flip probability of this noise process was set to $p' = 0.00110$ to match the single-qubit gate error rate reported in [23]. To each two-qubit gate in the recovery circuit, we associated a corresponding pair of parallel single-qubit phase-flip noise processes. This phase flip probability was set to $p'' = 0.00113$, as computed from the two-qubit gate error rate of [23].

Taking the product of these sequences of phase-flip noise processes, we compute that, throughout the recovery circuit, each qubit incurs a probabilistic phase-error rate of $p = 0.091$. The global noise process $\mathcal{N}$, then, consists of three independent PD channels, each with error probability $p = 0.091$.

In the case where a single physical qubit is used as a quantum memory without error correction, the error rate will be smaller, as the noisy gates are not present. The error model in this case is

single-qubit phase damping with $T_2 = 19\ \mu s$. The performance of the no-encoding case is fairly compared to the error correction case by choosing the duration of this phase-damping decay to match the duration of the recovery circuit. Taking the single-qubit and two-qubit gate times to be 10 *ns* and 40 *ns*, respectively, as reported in[23], and taking the specifications of the variational recovery circuit described in Appendix 6.6.2, the recovery duration is computed to be 1.8 μs. Accordingly, the no-encoding physical qubit noise process over this time step is probabilistic phase damping with error probability $p = 0.045$.

In the second simulation, we considered a setting where no active recovery is available and APD noise acts continuously during the wait time between encoding and decoding. We chose the value $T_2 = 19\ \mu s$ to match the dephasing rates reported in[23]. Although the T_1 times in this work were roughly twice that of T_2, we consider a model in which $T_1 = 57\ \mu s$, three times that of T_2. We choose this regime so that the error model differs sufficiently from isotropic depolarizing noise, in which the five-qubit code is known to be optimal[273]. In this regime, the dephasing errors are more dominant than the amplitude damping errors (which are coherent combinations of Pauli X and Y errors).

6.6.2 Numerical simulation of QVECTOR

We simulated the QVECTOR protocol using a Python script supplemented with the QuTiP library[161,162]. Since our simulations involve a single logical qubit, average fidelities were computed over the one qubit stabilizer states $|s_i\rangle$: $|+\rangle$, $|-\rangle$, $|0\rangle$, $|1\rangle$, $|+i\rangle$ and $|-i\rangle$, which form a 2-design. For systems with 2 or more logical qubits, we could use an approximate 2-design circuit such as the one described in[239] (See Appendix 6.6.3).

The simulation of the QVECTOR protocol comprises: 1) preparation of the initial state for the code register $\rho_{s_i} = |s_i\rangle\langle s_i| \otimes |0^{(n-k)}\rangle\langle 0^{(n-k)}|$, 2) application of the encoding channel $\mathcal{V}_{\vec{p}}$, 3) application of the noise process $\mathcal{N}$, 4) application of the recovery channel $\mathcal{W}_{\vec{q}}$ with r refresh qubits

and 5) decoding by application of $\mathcal{V}_{\vec{p}}^{\dagger}$. The final average fidelity is computed as:

$$\langle F \rangle = \frac{1}{N_s} \sum_{i=1}^{N_s} \mathrm{F}\left(\rho_{s_i}, \mathcal{V}_{\vec{p}}^{\dagger} \mathrm{Tr}_R\left[\mathcal{W}_{\vec{q}}\left(\mathcal{N}\mathcal{V}_{\vec{p}}\left(\rho_{s_i}\right) \otimes |0^{(r)}\rangle \langle 0^{(r)}|\right)\right]\right) \tag{6.10}$$

where N_s stands for the number of states employed to compute the average fidelity and $\mathrm{F}(\rho,\sigma) = \mathrm{tr}(\sqrt{\sigma^{1/2}\rho\sigma^{1/2}})^2$. The partial trace is performed over the refresh qubits used for recovery, which are initialized in the state $|0^{(r)}\rangle$. In the simulations presented in this work $N_s = 6$, corresponding to state 2-design formed by the six one-qubit stabilizer states. For our example of the phase damping channel, $n = 3$, $k = 2$ and $r = 2$. In our example of APD, we explored an approach without recovery, such that $n = 5$, $k = 4$, and $r = 0$.

The encoding $\mathcal{V}_{\vec{p}}$ and recovery $\mathcal{W}_{\vec{q}}$ were implemented using programmable circuits[83,319]. These types of circuits generally comprise a fixed networks of gates, where the parameters associated to the gates, i.e. rotation angles, constitute the variables for optimization. The pattern defining the network of gates is regarded as the unit-cell, which can be repeated to increase the flexibility of the model. The programmable circuits employed in this work are illustrated in Figure 6.4. The unit-cells of these circuits follow a pattern consisting of layers of single qubit rotations interleaved with entangling blocks. This heuristic construction is very amenable to current quantum hardware[168,234].

Our first programmable circuits, shown in Figure 6.4(a), are comprised of two layers of single-qubit rotations and two entangling layers. The layers of single-qubit rotations contain X_θ and Z_θ rotations applied on each qubit, where the notation A_θ stands for $\exp(-i\frac{\theta}{2}A)$. The entangling layers comprise all the possible adjacent controlled-Z_θ gates, with controls in the odd (even) qubits, and targets in the even (odd) qubits. The total number of parameters in this circuit is $2n + l(5n-1)$ where l is the number of repetitions of the unit cell. The unit-cell of our second programmable circuit, shown in Figure 6.4(b), comprises layers of single qubit arbitrary rotations

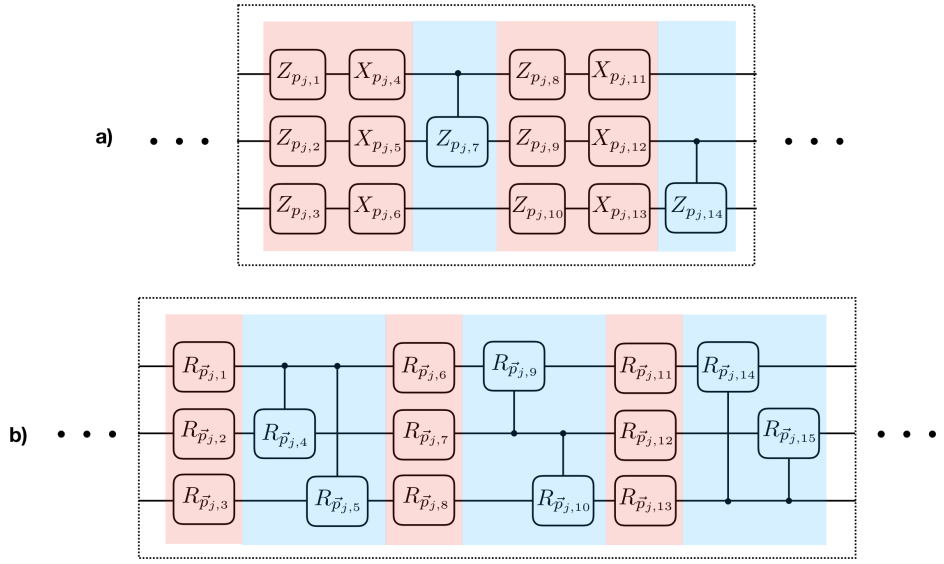


Figure 6.4: Unit-cells of programmable circuits employed for QVECTOR simulations for a three-qubit register. Both circuits comprise alternating layers of single-qubit rotations (red) and entangling operations (blue). a) Unit-cell of adjacent controlled-Z_θ rotations interleaved with layers of Z_θ and X_θ rotations, where $Z_\theta = \exp(-i\frac{\theta}{2}Z)$ (Analogous for X). b) Unit-cell comprises all the possible controlled-arbitrary single-qubit rotations (denoted by $R_{\vec{p}}$) in a given register, interleaved with layers of arbitrary single qubit rotations. Here $R_{\vec{p}} = Z_{p_1}X_{p_2}Z_{p_3}$. The index j runs from 1 to l, where l is the number of repetitions of the unit-cell. All the circuits are complemented with a single-qubit rotation layer after the last repetition.

interleaved with controlled- single qubit arbitrary rotations from the i-th qubit to the rest of the qubits in the register. Arbitrary single-qubit rotations were implemented as $R_{\vec{p}} = Z_{p_1}X_{p_2}Z_{p_3}$. The total number of parameters of this circuit is $3ln(2n-1)+3n$. In the simulations of the phase damping channel, we represented the circuit for $\mathcal{V}_{\vec{p}}$ an $\mathcal{W}_{\vec{q}}$ using 10 and 15 repetitions of the circuit of Figure 6.4(a). For the encoder of the APD example, we employed three repetitions of the unit-cell of Figure 6.4(b).

After determining the form of the unitaries for $\mathcal{V}_{\vec{p}}$ and $\mathcal{W}_{\vec{q}}$, the QVECTOR simulation proceeds by optimizing the fidelity in Eq. 6.10. For our numerical simulations, we employed the L-BFGS method[55] with a numerical gradient (central finite difference formula with step size $h = 10^{-6}$). The circuit parameters were initialized by generating 100 random parameter settings, drawn uniformly

from the range $[0, 4\pi)$, and selecting the set with the highest fidelity. Several optimizations were launched in parallel and the best result was selected. Our numerical explorations indicated that the average fidelity cost function might contain several local optima, and sampling different initial points for the optimization might benefit the success of the procedure.

We point out that in experimental implementations of QVECTOR, the fidelity cost function will be affected by errors introduced by sampling, as well as SPAM errors. In this scenario, the procedure might benefit from the use of optimization algorithms more tolerant to noise, such as Simultaneous Perturbation Stochastic Approximation (SPSA)[234], as well as from global optimization techniques such as Basin-Hopping[347]. Additionally, we expect that, in order to more-effectively estimate the average fidelity in the presence of these unwanted fluctuations, randomized benchmarking techniques should be employed (as outlined in the outlook section).

Lastly, we describe the bi-convex optimization routine which was used to compute the optimal average fidelities for the APD noise processes at different wait times. This method developed in[184] takes advantage of the fact that the average fidelity metric is a bi-linear, and, therefore, bi-convex function of the encoding and decoding channels,

$$\begin{aligned}
&\langle F(p\mathcal{D}+(1-p)\mathcal{D}',\mathcal{N},q\mathcal{E}+(1-q)\mathcal{E}')\rangle \geq p\langle F(\mathcal{D},\mathcal{N},q\mathcal{E}+(1-q)\mathcal{E}')\rangle+(1-p)\langle F(\mathcal{D}',\mathcal{N},q\mathcal{E}+(1-q)\mathcal{E}')\rangle \\
&\langle F(p\mathcal{D}+(1-p)\mathcal{D}',\mathcal{N},q\mathcal{E}+(1-q)\mathcal{E}')\rangle \geq q\langle F(p\mathcal{D}+(1-p)\mathcal{D}',\mathcal{N},\mathcal{E})\rangle+(1-q)\langle F(p\mathcal{D}+(1-p)\mathcal{D}',\mathcal{N},\mathcal{E}')\rangle,
\end{aligned} \tag{6.11}$$

where, although these hold with equalities, the inequalities suffice to enable the convex optimization method. The method proceeds by first choosing a random initial encoding $\mathcal{E}$ (chosen as an isometry), and then performing semidefinite programming to optimize the average fidelity with respect to the decoding $\mathcal{D}$, which is implemented using CVX, a package for specifying and solving convex programs[124,125]. Then, setting the decoding to this optimized variable, the average fidelity is convex-optimized with respect to the encoding map $\mathcal{E}$. This process is iterated, with

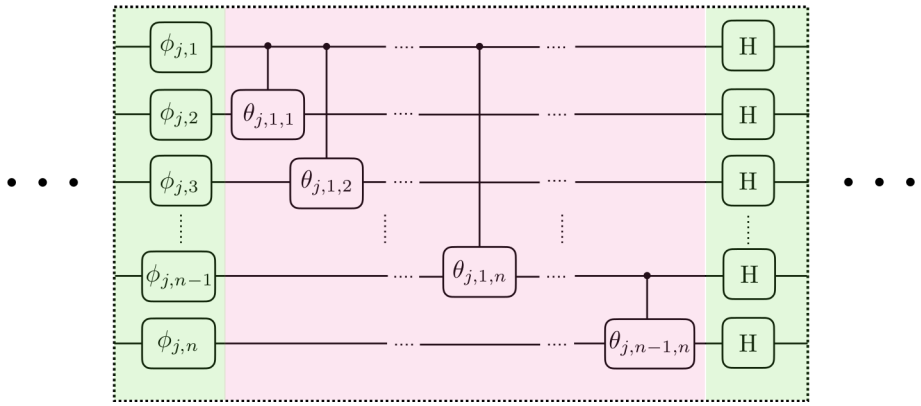


Figure 6.5: Unit-cell of the quantum circuit that implements an approximate unitary 2-design according to[239] in an n-qubit register. Layers are separated by colors. The one- and two-qubit gates in the first two layers correspond to diag(1, $e^{i\phi_{j,p}}$) and diag(1, 1, 1, $e^{i\theta_{j,p,q}}$), respectively. The phases $\phi_{j,p}$ and $\theta_{j,p,q}$ are chosen from $\{0, 2\pi/3, 4\pi/3\}$ and $\{0, \pi\}$, respectively, uniformly at random. The third layer comprises Hadamard gates (H) applied to all the qubits. All the gates in the first and the second layers are diagonal in the Pauli-Z basis and can be applied simultaneously. The index j runs from 1 to ℓ, where ℓ is the number of repetitions of the unit-cell.

average fidelity increasing in each step until the improvement in a step falls below a chosen threshold value. In practice, the optimized average fidelity varies from one run to the next depending on the initial encodings (the procedure is, otherwise, deterministic). To improve confidence that the obtained vale is sufficiently close to the optimal value, we perform many runs, decreasing the threshold value until the obtained average fidelities become sufficiently independent of the choice of threshold. Accordingly, although we can build substantial evidence for the value being close to optimal, this method can only definitively obtain a lower bound on the optimal average fidelity.

6.6.3 Accuracy of average fidelity estimate from approximate unitary 2-design

First we prove that the approximate 2-design of[239] (See Figure 6.5), with ℓ applications of the randomization circuit, leads to an estimator of the true average fidelity with bias upper bounded by $\frac{2^{k(\ell+1)}+2^{k\ell}-2}{2^{2k\ell}(2^k-1)} \sim \mathcal{O}(1/2^{k\ell})$. Let ν_ℓ be the measure on the unitary group that is sampled from using the approximate 2-design of[239] with ℓ repetitions. In the average fidelity estimation scheme, if we

draw from ν_ℓ instead of an exact 2-design (such as the Haar measure), we will be sampling from a biased estimator with mean

$$\langle E_\ell \rangle = \int \langle 0| U^\dagger \mathcal{M} \left(U |0\rangle \langle 0| U^\dagger \right) U |0\rangle \, d\nu_\ell(U), \tag{6.12}$$

where $\mathcal{M} = \mathcal{V}_{\vec{p}}^\dagger \mathcal{W}_{\vec{q}} \mathcal{V}_{\vec{p}}$. We give an upper bound on the bias $|\langle F \rangle - \langle E_\ell \rangle|$. The 2-design average over this measure may be renormalized in order to interpret it as a quantum channel $\mathcal{R}_\ell(\sigma) = d \int U^{\otimes 2} \sigma U^{\otimes 2^\dagger} d\nu_\ell(U)$. As shown in[239], this quantum channel can be written as a convex combination of two quantum channels,

$$\mathcal{R}_\ell(\sigma) = (1 - p_\ell)\mathcal{G}(\sigma) + p_\ell \mathcal{C}_\ell(\sigma), \tag{6.13}$$

where $\mathcal{G}$ is the renormalized average of an exact 2-design, $\mathcal{C}_\ell$ is a quantum channel, and $p_\ell = \frac{d^{\ell+1}+d^\ell-2}{d^{2\ell}(d-1)}$, with $d = 2^k$ being the Hilbert space dimension. To leverage this quantum channel interpretation of the 2-design, we rewrite the expression in Equation 6.12 as follows,

$$\langle E_\ell \rangle = \int \mathrm{Tr}\left[\mathcal{M} \otimes \mathcal{I} \left(U \otimes U |00\rangle \langle 00| U^\dagger \otimes U^\dagger \right) \mathbb{F} \right] d\nu_\ell(U) \tag{6.14}$$

$$= \frac{1}{d} \mathrm{Tr}\left[\mathcal{M} \otimes \mathcal{I} \left(\mathcal{R}_\ell(|00\rangle \langle 00|) \right) \mathbb{F} \right], \tag{6.15}$$

where $\mathbb{F}$ is the swap operator on the two systems. Replacing the channel $\mathcal{R}_\ell$ with the convex combination in Equation 6.13, we obtain an expression for the estimator mean in terms of the

actual mean,

$$\langle E_\ell \rangle = \frac{1}{d}(1-p_\ell)\mathrm{Tr}\left[\mathcal{M}\otimes\mathcal{I}\left(\mathcal{G}(|00\rangle\langle 00|)\right)\mathbb{F}\right] \tag{6.16}$$

$$+\frac{1}{d}p_\ell\mathrm{Tr}\left[\mathcal{M}\otimes\mathcal{I}\left(\mathcal{C}_\ell(|00\rangle\langle 00|)\right)\mathbb{F}\right] \tag{6.17}$$

$$=(1-p_\ell)\langle F\rangle + p_\ell\delta_\ell. \tag{6.18}$$

The bias of the estimator is $|\langle F\rangle - \langle E_\ell\rangle| = p_\ell|\langle F\rangle - \delta_\ell|$. To bound this value, we bound

$$\delta_\ell = \frac{1}{d}\mathrm{Tr}\left[\mathcal{M}\otimes\mathcal{I}\left(\mathcal{C}_\ell(|00\rangle\langle 00|)\right)\mathbb{F}\right]. \tag{6.19}$$

From Equation (12) in[239],

$$\mathcal{C}_\ell(|00\rangle\langle 00|) = \alpha P_{\mathrm{sym}} + \beta\sum_i |ii\rangle\langle ii|, \tag{6.20}$$

where $\alpha,\beta\geq 0$ and P_{sym} is the projector into the symmetric subspace. Since $\mathcal{C}_\ell$ is separable, it is invariant under partial transpose of either system. The partial transpose of the swap operator is the unnormalized Bell state $\mathbb{F}^{T_B} = \sum_{i,j}|ii\rangle\langle jj| \equiv d\Omega$. Since the trace of a bipartite operator is equal to the trace of the partial transpose of that bipartite operator, we can use $\mathcal{C}_\ell(|00\rangle\langle 00|)^{T_B} = \mathcal{C}_\ell(|00\rangle\langle 00|)$ and $\frac{1}{d}\mathbb{F}^{T_B} = \Omega$ to obtain

$$\delta_\ell = \mathrm{Tr}\left[\mathcal{M}\otimes\mathcal{I}\left(\mathcal{C}_\ell(|00\rangle\langle 00|)\right)\Omega\right]. \tag{6.21}$$

Observing that $\mathrm{Tr}\left[\mathcal{M}\otimes\mathcal{I}\left(\mathcal{C}_\ell(|00\rangle\langle 00|)\right)\Omega\right]$ is the inner product of two quantum states, we can upper bound this value by 1. Thus, the bias of the estimator is upper bounded by $p_\ell = \frac{d^{\ell+1}+d^\ell-2}{d^{2\ell}(d-1)} \sim \mathcal{O}(1/d^\ell)$.

After N samples from this biased estimator, our estimated average fidelity value is expected to

deviate from the estimator mean $\langle E_\ell \rangle$ by $\sqrt{\langle E_\ell \rangle (1 - \langle E_\ell \rangle)/N}$. An upper bound on the expected deviation of the sampling-estimated average fidelity from the true average fidelity is

$$\frac{1}{\sqrt{N}} + \frac{d^{\ell+1} + d^\ell - 2}{d^{2\ell}(d-1)} \sim \mathcal{O}\left(\frac{1}{\sqrt{N}} + \frac{1}{d^\ell}\right). \tag{6.22}$$

7

Generative adversarial quantum machine learning for continuous distributions

Apart from minor modifications, this chapter originally appeared as:

"Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions". Jonathan Romero and Alán Aspuru-Guzik, arXiv:1901.00848 (2018). Reproduced with permission from the authors.

7.1 Introduction

Quantum computing, a technology that relies on the properties of quantum systems to process information, is rapidly reaching maturity. Important problems that are hard to solve on classical computers based on transistors, such as factoring and simulating quantum systems, can be solved more efficiently using quantum computers[210,308]. These devices are nearing the noisy intermediate-scale quantum (NISQ) era[269], corresponding to machines with 50 to 100 qubits and capable of executing circuits with depths on the order of thousands of elementary two-qubit operations[269,356]. While NISQ devices will not be able to implement error-correction, as opposed to fault-tolerant quantum computers (FTQC), they are expected to provide computational advantages over classical supercomputers for certain problems, which includes sampling from hard-to-simulate probability distributions[1,43,269].

The limitations in the number of qubits and coherence times of NISQ devices have encouraged

the adoption of the hybrid quantum-classical (HQC) framework as the *de facto* strategy to design practical algorithms in the near term. The basic idea behind the HQC framework is that a computational problem can be divided into several subtasks, several of which can be executed more efficiently using a quantum computer while the rest can be deployed to a classical computer. A subset of HQC algorithms called adaptive hybrid quantum-classical (AHQC) algorithms, use classical resources to perform optimization of algorithm parameters. In this case, the quantum subtask generally refers to the process of preparing a parameterized quantum state, followed by the measurement of the expectation values of a polynomial number of observables that encode information relevant to the problem-of-interest. The parameterized quantum state is obtained using a *variational quantum circuit*, which consists of a set of tunable quantum gates whose parameters are subject to optimization. Examples of HQC algorithms include the variational quantum eigensolver (VQE)[226,255], the quantum approximate optimization algorithm (QAOA)[104], the variational quantum error-correction scheme (QVECTOR)[163], among others.

The HQC framework has also been adopted as the basis for designing quantum machine learning algorithms for NISQ devices. One of the first algorithms of this type is the quantum autoencoder (QAE)[280,350], where a variational quantum circuit is optimized to compress a set of quantum states. This is analogous to a classical autoencoder where an artificial neural network (ANN) is trained to compress classical datasets. The connection between neural networks and variational circuits has been further investigated, where it was shown that the HQC framework can approximate nonlinear functions just as classical neural networks can[60,231]. Furthermore, variational circuits have provided a new strategy for encoding classical information into quantum states, which is a fundamental step for machine learning applications. In contrast with *amplitude encoding*, in which the input vector is normalized and transformed directly into a quantum state, variational circuits can encode classical data by taking the input vector as the circuit parameters[138,231,297].

In recent months, the combination of the strategies described above for encoding classical data and designing HQC algorithms have led to an explosion of quantum machine algorithms for performing both discriminative[65,107,123,138,231,296,297,311] and generative tasks[32,153,254,343] on classical data using NISQ devices. In machine learning, discriminative models are trained to learn the conditional probability distribution of a target variable y with respect to a set of observations x, or $p(y|x)$. In contrast, generative models are trained to learn the joint probability distribution $p(x,y)$, or alternatively, the conditional probability of the observed data with respect to the target variable, $p(x|y)$. Most HQC algorithms for discriminative modeling use a variational quantum classifier[107,123,138,231,296,311], where a variational circuit is optimized to directly model $p(y|x)$ using training data $\{x_i, y_i\}$. Another strategy is to use a variational circuit as a quantum feature map for unsupervised classification with a support vector machine[138,297]. Meanwhile, HQC approaches to generative modeling have focused on modeling discrete probability distributions by using a variational circuit as a *Born machine*[32,209,312,372]. Born machines generate samples via projective measurement on the qubits, for example, by measuring the qubits in the computational basis. While this approach can learn probability distributions for small datasets used for benchmarking, such as Bars-and-Stripes[32,209], as well as quantum circuits for preparation of certain quantum states[32], the application of this model to general problems in generative modeling may be difficult due to the exponential scaling of the number of measurements required for sampling the distribution[32].

So far, HQC approaches for generative modeling of continuous probability distributions have not been developed. Most industrial applications, such as image and sound processing fall into this category. In this chapter we present a rather simple variational circuit architecture designed to generate continuous probability distributions. This *variational quantum generator* (VQG) comprises two quantum circuit components: the first one consists of a parameterized quantum circuit used to encode a classical random variable to a quantum state. The second circuit

corresponds to a variational circuit whose parameters are optimized to mimic the target classical probability distribution. The output distribution is obtained by measuring the expectation values of a set of predefined operators, whose values can be post-processed using a classical function. This construction provides considerable flexibility in the design of the variational circuit, allowing to easily incorporate VQG into classical neural network architectures. Furthermore, we show that our VQG architecture can be trained using an adversarial learning approach [121,212] leveraging automatic differentiation [28,35,240] to perform gradient-based optimization. That is, our VQG architecture learns to generate samples from the data distribution based on feedback obtained from a discriminator model, which simultaneously learns to distinguish between the samples coming from the real data distribution and those produced by the generator. We show that VQG can be trained using a classical neural network as well as a variational quantum classifier as discriminators.

This chapter is organized as follows: Section 7.2 briefly describes generative learning using generative adversarial networks and summarizes some proposals for generative learning on quantum computers. Section 7.3 describes the VQG architecture, its implementation, cost analysis, and training process using adversarial learning. In Section 7.4 we provide a proof-of-principle implementation and numerical simulation of a VQG example and describe the main challenges for its implementation on NISQ devices. Section 7.5 offers some concluding remarks.

7.2 Background

7.2.1 Classical and quantum generative adversarial learning (GANs)

The machine learning literature provides a variety of generative models. Most of them are trained using the principle of maximum likelihood, that consists of taking several samples from the data generating distribution to form a training set and changing the parameters of the model to maximize the likelihood of the observed data of being generated by the model. Generative models

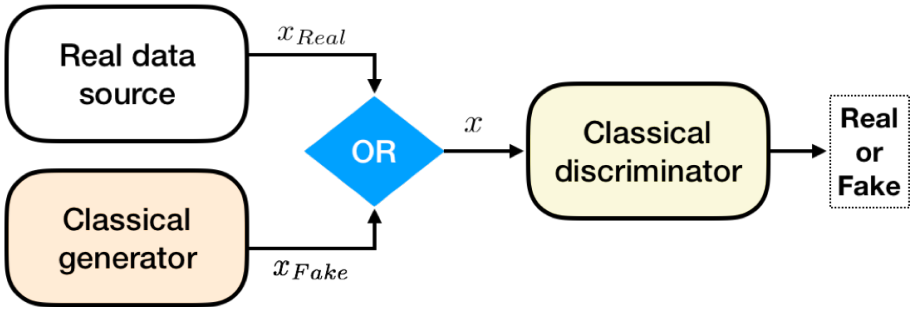


Figure 7.1: Depiction of the classical generative adversarial networks (GANs) scheme: the generator, equipped with random samples from a prior distribution (noise source), produces samples that attempt to mimic the real data samples. The discriminator outputs the probability that a given sample came from the real distribution rather than the synthetic one.

in machine learning can be classified as explicit or implicit, depending on whether or not a explicit form of the probability density function is used[119]. Very few tractable explicit models are known, and most of them rely on approximations to the density function. On the other hand, most of the implicit models consist of approximations that can mimic the process of sampling from the generating distribution. Implicit models are further classified into models that require several steps to generate a single sample, such as Markov chains, and models that can generate a sample in a single step. Generative adversarial networks (GANs) belong to the latter category.

Generally, GANs consist of two neural networks, the discriminator and the generator, competing against each other in a zero-sum game. Figure 7.1 illustrates the general framework of GANs. Given a prior distribution over the noise parameters $p_z(z)$, the generator consists of a neural network $F_G(z;\Theta_g)$ over the parameters Θ_g that generates the distribution p_g. On the other hand, the discriminator is another neural network $F_D(x;\Theta_d)$ that outputs a single scalar corresponding to the probability of x coming from the real data distribution. Accordingly, F_D is trained to maximize the probability of assigning the correct label to both the training examples and examples coming from F_G. Simultaneously, F_G is trained to minimize $log(1-F_D(F_G(z)))$, related to the probability of

fooling the discriminator. In summary, F_D and F_G play the following adversarial game:

$$\min_G \max_D (\mathbb{E}_{x\sim p_{data}(x)}[\log F_D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - F_D(F_G(z)))]). \tag{7.1}$$

Assuming that the discriminator and the generator have infinite capacity, meaning that they can represent any probability distribution, it is possible to show that the final stage of the game reaches a Nash equilibrium where the generator produces data that corresponds to the observed probability distribution, and the discriminator has 1/2 probability of discriminating correctly. Therefore, the final result of the GAN is a generator model that produces samples from the observed distribution by sampling the prior distribution $p_z(z)$. The space of z is usually called the latent space, and F_G is said to map samples from the latent space to the output space x. The adversarial framework has proven very successful at training the generator to model a variety of probability distributions, leading to practical applications in many fields, including image synthesis, semantic image editing, molecular discovery, among others[74,118,271]. Nowadays, the application of GANs constitute an exciting and fast growing research field that promises to impact many industries such as self-driving cars, finance, and drug and materials discovery[15,212].

Recently, different quantum adaptations of the GAN scheme have been proposed[77,151,212,312,372]. These methodologies can be characterized according to whether the data source and the models used as discriminator and generator are classical or quantum. The different scenarios considered so far are summarized in Figure 7.2. In particular, Ref.[212] offers a theoretical perspective on three possible adversarial learning scenarios. The first of these settings corresponds to a purely quantum version of GANs, where the data distribution is a quantum source and the models correspond to quantum circuits. This proposal is further developed in[77], and experimentally demonstrated for a proof-of-principle quantum computation with a superconducting qubit architecture[151]. The second scenario considers a classical generator that is trying to produce quantum data at an exponential

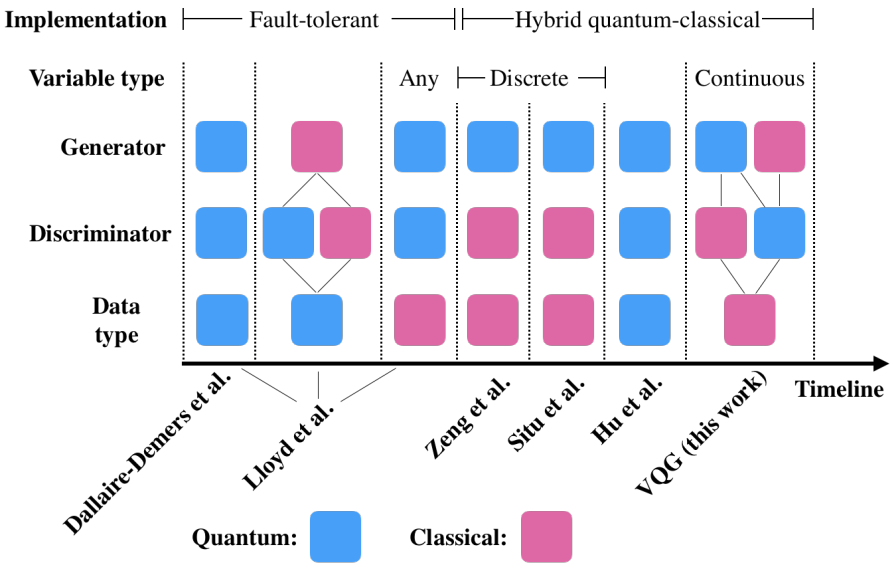


Figure 7.2: Timeline of the development of quantum generative adversarial network models (Dallaire-Demers et al.[77], Lloyd et al.[212], Zeng et al.[372], Situ et al.[312], Hu et al.[151] and this work (VQG)). We describe each proposal in terms of the nature of the data, the discriminator and the generator, that can be either classical or quantum. Lines indicate possible combinations of models and data type. For those models where the type of the data generated is classical, we describe whether the type of variable is discrete or continuous. We also describe the type of implementation proposed, whether it is based on a fault-tolerant model or a hybrid quantum-classical one.

cost. The third scenario corresponds to classical data encoded in the amplitudes of a quantum state, such as quantum generators and discriminator can be employed. As described in[212], these proposals are designed for error-corrected quantum computers. More recently, some groups have proposed hybrid-quantum classical adversarial learning schemes that could be implemented on NISQ devices. These approaches utilize a classical data source and a classical discriminator combined with a variational circuit sampled as a Born machine as generator[312,372]. As noted earlier, the Born machine approach consists of generating a discrete distribution via projective measurement on the qubits.

7.3 THE VARIATIONAL QUANTUM GENERATOR ARCHITECTURE

7.3.1 ARCHITECTURE

Existing quantum models for generative learning collect data by measuring the system as a Born machine, which is convenient for discrete distributions but can cannot be easily adapted for continuous cases. We propose a scheme to generate continuous distributions that builds on the principles of HQC computing. Consider a real data source that outputs observations of an unknown distribution, represented by the variable $x \in \mathbb{R}^N$. The purpose of our variational quantum generator is to produce classical samples x_{Fake} that mimic the observed distribution. To achieve this, we propose the construction depicted in Figure 7.3, that includes two variational circuits, a quantum encoding circuit $R(z)$ acting on r qubits and the generator circuit $G(\Theta_g)$ acting on n qubits with $n \geq r$.

The quantum encoding circuit, which we describe in detail in the next subsection, takes as input a classical random variable $z \sim p_z(z); z \in \mathbb{R}^O$ as a parameter and prepares the state $R(z)\,|0^{\otimes r}\rangle = |\phi(z)\rangle$. This is the equivalent to the random source employed in classical GANs, where the space of the variable z would correspond to the latent space in the language of generative models. Correspondingly, the manifold of states $\{|\phi(z)\rangle\}$ would constitute the quantum latent space. The second circuit, $G(\Theta_g)$ acts as the generator model, mapping from the latent manifold to the manifold of observed data x: $G(\Theta_g)\,|\phi(x)\rangle = |\psi(z,\Theta_g)\rangle$. To map this state to a classical value we employ a *measurement decoding* scheme, where the sample $P \in \mathcal{R}^M$ is generated by measuring the expectation value of a fixed set of observables expressed as strings of Pauli strings $\{P_i\}_{i=1,\cdots,M}$:

$$P = [\langle P_1\rangle, \langle P_2\rangle, \cdots, \langle P_M\rangle] \tag{7.2}$$

$$\text{where} \quad \langle P_i\rangle = \langle \psi(x;\Theta_g)|P_i|\psi(x;\Theta_g)\rangle. \tag{7.3}$$

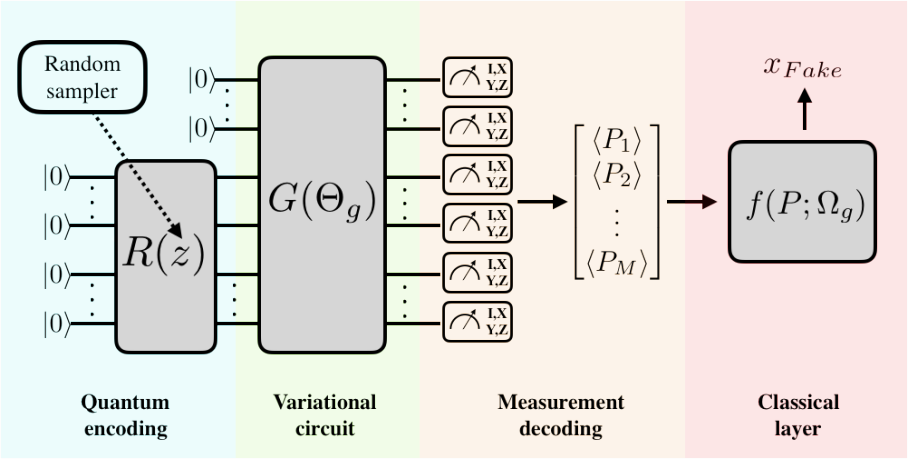


Figure 7.3: Circuit architecture of the proposed quantum generator, comprising a circuit that generates states from a latent space (z) using the variational circuit $G(\Theta_g)$. The random variable z is mapped to a quantum state using the quantum encoding circuit R. By measuring a fixed set of operators on the generated state, the quantum circuit produces a classical vector $P = [\langle P_1\rangle, \cdots, \langle P_M\rangle]$, that passes through a classical function $f(P;\Omega_g)$, to produce the fake sample x_{Fake}.

P is then transformed by a classical function to generate x_{Fake}:

$$x_{Fake} = f_g\left(P;\Omega_g\right), \tag{7.4}$$

$$\tag{7.5}$$

where Ω_g represents a vector of real parameters associated to the classical function. In what follows, we describe each of the components of VQG in greater detail.

7.3.1.1 Quantum encoding circuit

The process of encoding classical inputs in a quantum state can be interpreted as applying a nonlinear feature map that maps data to a quantum Hilbert space, a process also called *quantum feature map* or *quantum encoding*, as described by Schuld et al.[297]. The quantum circuit implementing this mapping on a digital quantum computer corresponds to the *quantum feature circuit* or *encoding circuit*. We distinguish between two classes of quantum encoding in this

chapter:

1. *Amplitude encoding*: In the first case, a vector $x \in \mathbb{R}^N$, corresponding to the data to be encoded, undergoes a transformation under a feature map: $\psi : \mathbb{R}^N \to \mathbb{C}^{2^n}$ that maps the information to a quantum state in n qubits. Since the length of the data vector is not necessarily a power of 2, the feature map might require some padding and appropriate normalization. Once the corresponding input state is obtained, we need to prepare this state on the quantum register of n qubits, $|\phi(x)\rangle$ using a preparation circuit S_x such that $S_x |0\rangle^{\otimes n} = |\psi(x)\rangle$.

2. *Variational encoding*: In this case, a fixed variational circuit $E(f_E(x))$ encodes the data by inputting the classical information as the circuit parameters. Here, f is a classical feature map: $f_E : \mathbb{R}^N \to \mathbb{R}^M$, such as the final input state is prepared as $E(f(x))\,|0\rangle^{\otimes n} = |\phi(x)\rangle$.

Notice that in amplitude encoding, the vector is mapped classically to a quantum state. Consequently, we need to find the corresponding circuit that prepares the state to a desired accuracy. This can be done using general purpose compilation routines for preparing general quantum states on quantum registers [127,244,259,315]. In the case of amplitude encoding, the number of qubits required scales as $O(\log(N))$ while the depth of the circuit for state preparation is $O(N)$ [259], with N being the size of the classical vector to be mapped. The number of gates required for state preparation of these circuits (In the order of thousands for ten qubits [244]) might constitute a challenge for NISQ devices.

In contrast, the variational encoding strategy encodes the classical vector as the parameters of a fixed variational circuit. This implies that the circuit layout employed for all the input vectors is the same, which simplifies compilation. It is also likely that the errors introduced by this encoding procedure are mostly systematic and therefore can be more easily mitigated. Most of the variational encodings proposed so far employ circuits with $O(N)$ qubits and only $O(1)$ circuit

depth[123,123,153,231,296,297], which makes encoding more amenable to NISQ devices at the cost of increasing requirement in the number of qubits compared to amplitude encoding.

Both amplitude encoding and variational encodings have been used in machine learning proposals for classification[38,107,138,231,296,297], and can be employed as part of the VQG architecture. In the space of variational encodings, some specific classes of circuits have been proposed. Some examples include *product encoding*, in which each element of the vector x is mapped to a one qubit state by a specific quantum circuit[123,153,296,297]. Other approaches incorporate more layers of single qubit gates whose parameters are given by the elements of the feature vector, followed by circuit blocks made out of fixed entangling operations[138]. A particular strategy that can be used to introduce non-linearity is the so-called *tensorial mapping*, which consists of preparing several copies of the quantum state encoding the data[231,297]. An example of a variational encoding combining product encoding and tensorial mapping is the following preparation circuit:

$$U(x) = \prod_{k}^{N} \prod_{i}^{n_k} R_Z^i(f(x_k)) R_Y^i(g(x_k)), \tag{7.6}$$

where each element of the vector x is mapped by a circuit acting on a fixed number of qubits, n_k, and f and g correspond to non-linear activation functions. The notation $R_V^I(\alpha) = e^{-i\frac{\alpha}{2}V_I}$ indicates a general rotation under the operator V acting on the set of qubits I. Notice that in the map of equation 7.6, non-linearities are introduced via the use of non-linear functions as part of the mapping and by application of the tensorial map.

7.3.1.2 VARIATIONAL CIRCUIT

$G(\Theta_g)$ plays the role of the variational circuit in our VQG architecture. Most variational circuits are designed to prepare strongly entangled quantum states. This follows the general intuition that the variational circuits employed should be able to map the input data into quantum states that are

hard to manipulate on classical computers. In addition, variational circuits must be able to spot different types of correlations in the input data, which requires circuits with the ability to explore Hilbert space sufficiently. Variational algorithms such as QAE and QVECTOR have been implemented using quantum circuits composed by a fixed networks of a polynomial number of gates, usually restricted to single-qubit and two-qubit operations, with angles that serve as variational parameters. The pattern defining the network of gates can be seen as a *unit-cell* or *circuit block* that can be repeated to increase the flexibility of the model. The term Multilayer Quantum Circuit (MPQC) has been recently coined to describe this type of variational circuit architecture[95]. MPQC circuits have been widely used as quantum models for classification tasks[138,296] and has been shown to generate discrete probability distributions that cannot be efficiently simulated by classical neural networks[95]. We describe the specific architecture of some of these circuits in Appendix 7.6.1.

Apart from MPQC circuits, it is also possible to use a circuit implementing the evolution under a family of Hamiltonians known to generate strongly correlated states. In this case, the coefficients of the Hamiltonian terms can be used as variational parameters. For instance, Mitarai et al.[231] used the evolution under a transverse Ising Hamiltonian to perform simulations of quantum classification and to model nonlinear functions using variational circuits[231]. The circuits implementing evolution under a given Hamiltonian may require Trotterization. In this case, each Trotter step might be interpreted as a circuit block, in analogy with the Hamiltonian variational approach described in Ref.[353].

7.3.1.3 Measurement decoding and post-processing

The process of measurement decoding generates samples from the generator by estimating the vector of expectation values $P = [\langle P_1 \rangle, \langle P_2 \rangle, \cdots, \langle P_M \rangle]$. The choice of operators for decoding depends on the problem at hand and constitutes a hyper-parameter of the model. The cost of

estimating the vector P with measurement averaging, assuming each operator is measured independently and with fixed precision ε, is $O\left(\sum_{i=1}^{M} \frac{Var[P_i]}{\varepsilon^2}\right)$. The associated measurement cost is not different from other HQC algorithms such as VQE, where the expectation value of the Hamiltonian is computed by a weighted average of the expectation value of a polynomial number of observables[226]. Here we assume that ε is independent of subsequent transformations of the vector P and is small enough to carry out the optimization and the generation of samples x successfully.

If the training of the VQG model is carried out with gradient-based optimization, this will require the estimation of $\nabla_{\Theta_G} P$. In this case, the total number of measurements employed depends on the number of circuit runs used per gradient estimation (N_{grad}) and the total number of gradient evaluations required by the optimization (N_{opt}). As described in Appendix 7.6.1, $N_{grad} \sim O(N_p/\varepsilon^2)$ where N_p is the total number of parameterized gates in the variational circuit. To generate the final sample x, the generator can incorporate a classical function that transforms the measurement vector, P. In general, we can express f_g as $f_g(x) = h(WP+b)$, where $W \in \mathbb{R}^{N\times M}$, $b \in \mathbb{R}^N$ and h is a function that can be nonlinear $h : \mathbb{R}^N \to \mathbb{R}^N$. To unify the notation, we designate $\Omega_g = (W, b)$ as in Figure 7.3. This construction makes VQG a hybrid quantum-classical architecture and therefore the evaluation of the model and its derivatives will require feedback between the classical computer and the quantum processor. We describe this process in more detail in the next section, where we discuss how to train the VQG model.

7.3.2 Training and cost function

The VQG architecture could be trained by direct maximization of the log-likelihood. However we have chosen to use an adversarial learning approach, which has certain advantages as described in Section 7.2.1. The adversarial setting requires a discriminator function $F_D(x;\Theta_d)$, parameterized by Θ_d, that receives the sample x as input and outputs an approximation to the probability of the

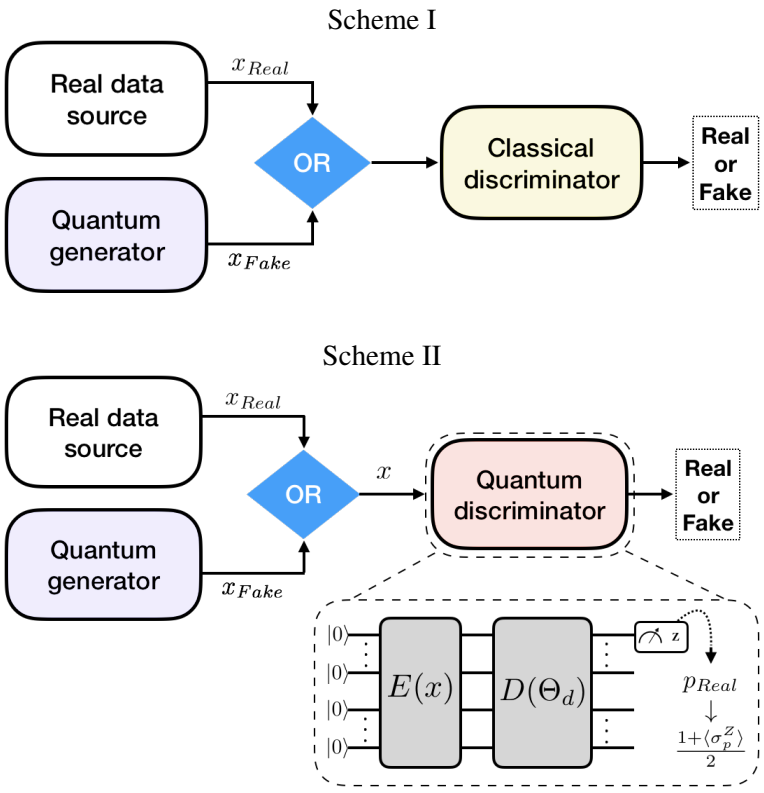


Figure 7.4: Two different schemes for training a quantum generator of classical data using an adversarial learning approach: the first scheme (Scheme I) employs a classical discriminator (e.g. classical neural network), whereas the second scheme employs a quantum discriminator (Scheme II), which consists of a quantum circuit that encodes the classical sample ($E(x)$) and a variational circuit ($D(\Theta_d)$), whose parameters are optimized such as the measured observable (σ_p^Z) describes the probability of the sample to come from the real distribution.

sample originating from the real distribution. We will describe the architecture of the discriminator shortly, but assuming we can compute the necessary gradients, this discriminator could be trained using the same cost function employed in classical GANs (Eq. 7.1). We rewrite this expression to make explicit the dependency in the parameters:

$$C_d(\Theta_d,\Theta_g) = -\frac{1}{2}\mathbb{E}_{x\sim p_{data}(x)}[\log F_D(x;\Theta_d)] - \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}\log[1-F_D((F_G(z;\Theta_g,\Omega_g);\Theta_d)], \quad (7.7)$$

where $F_G(z;\Theta_g,\Omega_g)$ is the function corresponding to VQG. The first term in Equation 7.7 corresponds to the probability of the discriminator to succeed at classifying data coming from the real source correctly, while the second term represents the probability of the discriminator to succeed at identifying the sample created by the generator as fake. Notice that the discriminator needs to be trained on two batches of data: one corresponding to real samples (for which the discriminator should output 1) and a second batch created by the generator (for which the discriminator should output 0). In classical GANs, the original choice of the cost function of the generator is just the negation of the cost function of the discriminator, such as $C_g(\Theta_d,\Theta_g) = -C_d(\Theta_d,\Theta_g)$, and therefore the final optimization consists of a minimax game:

$$\min_{\Theta_g}\max_{\Theta_d} C_d(\Theta_d,\Theta_g). \tag{7.8}$$

At each step of the optimization, the parameters of each player are optimized while the parameters of the other player are kept fixed. One alternative to the cost function in Equation 7.7 is to use the inverse of the discriminator cost function for the generator, such that C_g becomes:

$$C_g(\Theta_d,\Theta_g) = -\mathbb{E}_{z\sim p_z(z)}[\log F_D((F_G(z;\Theta_g,\Omega_g);\Theta_d))]. \tag{7.9}$$

In this case, the generator minimizes the probability of the discriminator of being correct. This proposal, although heuristically motivated, has demonstrated the ability to facilitate the training process in classical GANs[119].

As in the case of other quantum machine learning approaches, we propose to use gradient based methods for the optimization. Most recent numerical and experimental demonstrations for classifiers based on variational circuits have employed methods such as simultaneous perturbation stochastic approximation (SPSA)[320,321] and stochastic gradient descent (SGD)[120]. SPSA is based

on numerical gradients and has been already employed in experimental demonstrations of VQE[168] and QML algorithms for classification[138]. The difficulty with SPSA is that the number of measurements required increases substantially as the gradient vanishes. In this case, the heuristic cost function for the generator (Equation 7.9) might require fewer measurements as it prevents vanishing gradients. In contrast, algorithms such as SGD generally work with analytical gradients in the context of classical neural networks. Employing these algorithms for our VQG requires computing gradients with respect to the circuit parameters Θ_g, Ω_g and Θ_d.

In order to compute gradients for the discriminator we need to define its structure. Since the VQG is designed to generate classical data, it is possible to perform the training using both classical and quantum discriminators. These two possible schemes are pictorially described in Figure 7.4. In the first scheme, the classification is performed by a classical discriminator, for example, a classical feed-forward neural network. Consequently, the discriminator can be trained by maximizing the cost function (Eq. 7.7) using standard back-propagation techniques for feed-forward neural networks.

In the second scheme, the classification is performed by quantum discriminator model, for example, a variational quantum classifier (VQC)[138,231,296,311]. As the input data is classical, the quantum discriminator comprises a quantum encoding circuit, $E(x)$, that maps the data point x to a quantum state, and a variational circuit D, with parameters Θ_d. A set of measurements provide the final values indicating the classification. Correspondingly, the structure of this quantum discriminator becomes analogous to the structure of the quantum generator, with the difference that the classical output produced by measurement needs to be transformed such that it corresponds to a probability distribution instead of an arbitrary vector. For the GAN implementation, the discriminator performs only binary classification and therefore $p(y|x)$ can be modeled by measuring a single observable, e. g. σ_a^Z, with a being the index of the designated qubit.

Correspondingly, the probability of x coming from the true distribution of the data can be estimated simply as $p_{Real} = \frac{1+\langle\sigma_p^Z\rangle_D}{2}$, where:

$$\langle\sigma_p^Z\rangle_D = \langle 0|E^\dagger(x)D^\dagger(\Theta_d)\sigma_p^Z D(\Theta_d)E(x)|0\rangle. \tag{7.10}$$

Consequently the gradient of the discriminator takes the form:

$$\frac{\partial F_D(x;\Theta_i^d)}{\partial\theta_i^d} = \frac{1}{2}\frac{\partial\langle 0|E^\dagger(x)D^\dagger(\Theta_d)\sigma_p^Z D(\Theta_d)E(x)|0\rangle}{\partial\theta_i^d}, \tag{7.11}$$

Eq. 7.11 can be evaluated using the standard techniques for computing gradients in variational circuits described in Appendix 7.6.2. On the other hand, training the generator implies computing gradients of both Eq. 7.8 and Eq. 7.9 with respect to the generator parameters, which ultimately requires the calculation of the following derivatives:

$$\frac{\partial F_D(F_G(z;\Theta_g,\Omega_g);\Theta_d)}{\partial\theta_g^i} = \sum_l\sum_k \frac{\partial F_D(x_l)}{\partial x_l}\frac{\partial x_l}{\partial\langle P_k\rangle}\frac{\partial\langle P_k\rangle}{\partial\theta_g^i}, \tag{7.12}$$

$$\frac{\partial F_D(F_G(z;\Theta_g,\Omega_g);\Theta_d)}{\partial\omega_g^i} = \sum_l \frac{\partial F_D(x_l)}{\partial x_l}\frac{\partial x_l}{\partial\omega_g^i}, \tag{7.13}$$

where we have used the following notation:

$$\Theta_d = [\theta_d^1, \theta_d^2, \cdots, \theta_d^{|\Theta_d|}]; \tag{7.14}$$

$$\Theta_g = [\theta_g^1, \theta_g^2, \cdots, \theta_g^{|\Theta_g|}]; \tag{7.15}$$

$$\Omega_g = [\omega_g^1, \omega_g^2, \cdots, \omega_g^{|\Omega_g|}]; x \qquad = [x_1, x_2, \cdots, x_N]. \tag{7.16}$$

Notice that the partial derivatives appearing in Eq. 7.12 are estimated differently depending on the type of discriminator used. In scheme I, $\frac{\partial F_D(x_l)}{\partial x_l}$, $\frac{\partial x_l}{\partial \omega_g^i}$ and $\frac{\partial F_D(x_l)}{\partial x_l}$ correspond to derivatives of classical functions and can be computed using established backpropagation techniques. In this case, only $\frac{\partial \langle P_k \rangle}{\partial \theta_g^i}$ corresponds to a derivative of a variational circuit. In contrast, $\frac{\partial F_D(x_l)}{\partial x_l}$ is also a derivative of a variational circuit in the case of Scheme II:

$$\frac{\partial F_D(x;\Theta_i^d)}{\partial x_i} = \frac{1}{2} \frac{\partial \langle 0|E^\dagger(x) D^\dagger(\Theta_d) \sigma_p^Z D(\Theta_d) E(x)|0\rangle}{\partial x_i}, \tag{7.17}$$

which implies taking derivatives of the encoding circuit or the corresponding encoding scheme. In case of variational encodings, the same techniques applied to compute the gradients of circuits G and D can be employed for computing Eq. 7.17. If the encoding involves pre-processing x with a classical function, the calculation of the gradient requires further unfolding as with Eq. 7.12.

In summary, to train the VQG model using adversarial learning, we need to compute gradients of variational circuits and apply backpropagation for classical functions. We review the calculation of existing techniques for computing analytical gradients of variational circuit in Appendix 7.6.2. To compute gradients of classical functions, we exploit state of the art automatic differentiation (AD) techniques[95,240]. AD is an algorithmic strategy to extend a program that computes numerical values of a function such as it can also compute arbitrary derivatives of the same function, as described in Appendix 7.6.3. This technique is widely used in machine learning to perform automatic calculation of derivatives for gradient-based optimization of models such as neural networks. AD also offers a convenient framework to propagate gradients between classical and quantum functions, as described in Ref.[35,295].

Figure 7.5 illustrates the calculation of gradients for a hybrid-quantum classical function, such

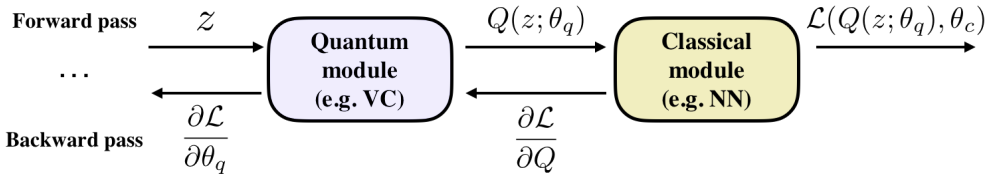


Figure 7.5: Application of reverse accumulation (See Appendix 7.6.2) for automatic differentiation of a hybrid quantum-classical architecture. The quantum module implements a function computed from a variational circuit (VC), $Q(x,\theta_q)$, as well as derivatives of this function with respect to x and θ_q. The classical module implements a classical function e.g. a neural network (NN), $\mathcal{L}(y,\theta_c)$, and its derivatives with respect to y and θ_c. In the forward pass, $\mathcal{L}(Q,\theta_c)$ is calculated by first computing $Q(z,\theta_q)$ using the quantum module and passing its value to the classical module. In the backward pass, the classical module computes $\frac{\partial \mathcal{L}}{\partial Q}$ and passes this information to the quantum module. The quantum module estimates $\frac{\partial Q}{\partial \theta_q}$ using the quantum processor and computes $\frac{\partial \mathcal{L}}{\partial \theta_q}$ by application of the chain rule: $\frac{\partial \mathcal{L}}{\partial \theta_q} = \frac{\partial \mathcal{L}}{\partial Q}\frac{\partial Q}{\partial \theta_q}$.

as a VQG module, using AD. The example shows two functions: the first one corresponds to the output of a variational circuit, $Q(z,\theta_q)$, where z and θ_q are both classical inputs (e.g. classical information encoded into the circuit and variational parameters, respectively). The second function is classical (e.g. a feed-forward neural network), taking as inputs the value of the function $Q(x,\theta_q)$ and parameters θ_c and producing the output $\mathcal{L}(Q,\theta_c)$. At the computational level, these functions are implemented as programming functions or instances of a computational class and are executed separately by quantum and classical modules, respectively. The quantum module can be interpreted as a classical computer that has access to a quantum processor, while the classical module incorporates only classical computing resources.

Suppose we want to compute $\frac{\partial \mathcal{L}}{\partial \theta_q}$ using AD. In the forward pass, $\mathcal{L}(Q,\theta_c)$ is calculated by first computing $Q(z,\theta_q)$ using the quantum module. This value is passed onto the classical module, which computes the final output given some value of θ_c. In the backward pass, the classical module estimates $\frac{\partial \mathcal{L}}{\partial Q}$ and passes this information backwards to the quantum module. The quantum module estimates $\frac{\partial Q}{\partial \theta_q}$ using the techniques described in Appendix 7.6.2 and uses the value of $\frac{\partial \mathcal{L}}{\partial Q}$ provided by the classical module to compute $\frac{\partial \mathcal{L}}{\partial \theta_q}$ by application of the chain rule: $\frac{\partial \mathcal{L}}{\partial \theta_q} = \frac{\partial \mathcal{L}}{\partial Q}\frac{\partial Q}{\partial \theta_q}$. An analogous procedure can be applied to compute all the derivatives required for training the VQG

Result: Optimal Θ_g, Ω_g and Θ_d
Data: N_s, N_e, S_d, S_g, Initial Θ_g, Ω_g and Θ_d;
for $n := 1$ *to* N_e **do**
  for $s_1 := 1$ *to* S_d **do**
    Sample N_s times from $p_z(z)$: $\{z^{(1)}, z^{(2)}, \cdots, z^{(N_s)}\}$;
    Sample N_s times from the data distribution: $\{x^{(1)}, x^{(2)}, \cdots, x^{(N_s)}\}$;
    Update Θ_d by ascending discriminator's gradient: $\nabla_{\Theta_d} \frac{1}{M} \sum_i^M C_d(z^{(i)}, x^{(i)}, \Theta_g, \Omega_g, \Theta_d)$;
  end
  for $s_2 := 1$ *to* S_g **do**
    Sample N_s times from $p_z(z)$: $\{z^{(1)}, z^{(2)}, \cdots, z^{(N_s)}\}$;
    Update Θ_g and Ω_g by descending generator's gradient: $\nabla_{\Theta_d} \frac{1}{N_s} \sum_i^{N_s} C_g(z^{(i)}, \Theta_g, \Omega_g, \Theta_d)$;
  end
end

ALGORITHM 1. Adversarial learning of a variational quantum generator (VQG). Training proceeds for N_e epochs. At each epoch, the parameters of the discriminator and the generator are updated separately, S_d and S_g times respectively. Cost functions are estimated by taking N_s samples of the real and the synthetic data distributions.

architecture using adversarial learning (Eqs. 7.12-7.13). With this infrastructure in place, the optimization of all the parameters of the model can be performed using standard gradient-based optimizers such as Adam or SGD. Algorithm 1 summarizes the pseudocode for the adversarial learning of the VQG model.

7.4 IMPLEMENTATION

7.4.1 NUMERICAL SIMULATIONS

To illustrate the implementation of the VQG model and demonstrate its feasibility, we designed a controlled experiment where the real data source is generated by a VQG instance with the same structure as the generator. This guarantees that a solution to the learning problem exists, allowing us to focus on studying the convergence of the training process. This also facilitates the assessment of the success of the training process by directly comparing the two distributions. In our experiment, the adversarial learning process incorporates the following elements, illustrated in

Figure 7.6(a):

1. *Generator:* Our generator corresponds to a VQG model composed of a product encoding circuit with two layers of one qubit gates incorporating the tensorial mapping strategy to introduce non-linearities. The variational circuit of the generator is a two qubit circuit with a layer of single-qubit Y rotations followed by evolution under the operator XX, for a total of three parameters. The measurement decoding is performed with a single operator $[\sigma_Z^1]$ without classical post-processing. This generator produces a probability distribution $p_G(x)$, with $x \in \mathbb{R}, \quad x \in [-1,1]$. At the beginning of the training, the variational circuit is initialized with parameters $\Theta_g = [2.3, 2.3, 1.0]$.

2. *Discriminator:* We tested the two schemes described in Figure 7.4 for training the generator. In scheme I, we employed the classical feed-forward neural network described in Figure 7.6(a) as discriminator. In scheme II, we used a quantum discriminator comprising a product encoding circuit and a variational circuit on three qubits. The variational circuit for the discriminator corresponds to a single $B(3,1)$ block encompassing a layer of arbitrary single qubit rotations, followed by parameterized C-Phase gates and finally a layer of single qubit X rotations. This type of variational circuit has been used in combination with amplitude encoding for classification tasks such as MNIST[296]. As in many application of classical GANs, we chose the discriminators to be more complex than the generator, having in this case more qubits, which is equivalent to a bigger size of the hidden layer.

3. *Real data source*: To generate the real data distribution, we employed a VQG model with the same structure as the one used in the generator and parameters fixed to $\Theta_g = [2.48, 2.52, 2.0]$. This corresponds to the classical univariate probability distribution, $p_{Data}(x), x \in \mathbb{R}, \quad x \in [-1,1]$, shown in Figure 7.6(a).

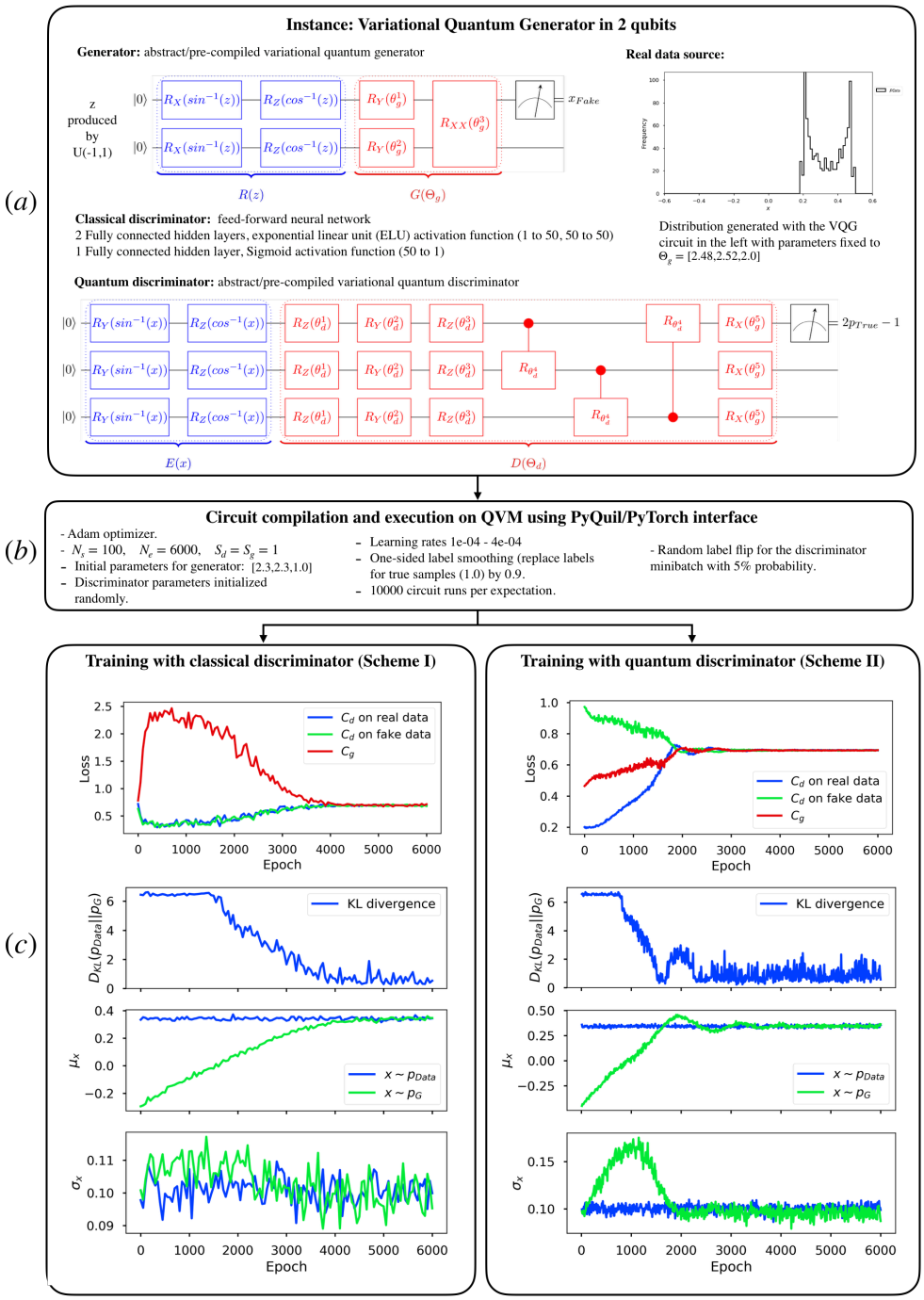


Figure 7.6: Example of the implementation and training of a VQG instance following the `algo2qpu` approach[311]. (a) Architecture of the generator and discriminators used in the numerical experiments. The part of the circuits corresponding to encoding circuits and variational circuits are shown in blue and red, respectively. The real distribution corresponds to the quantum generator architecture with parameters fixed at $\Theta_g = [2.48, 2.52, 2.0]$ and Pauli set $[\sigma_X^1]$. (b) Details of the implementation and execution of the experiments. We performed noiseless simulations using a QVM. The generator is initialized at $\Theta_g = [2.3, 2.3, 1.0]$. (c) Training dynamics using schemes I (left panel) and II (right panel). Each panel shows from top to bottom: loss functions (C_d and C_g) as a function of the number of epochs, Kullback-Leibler (KL) divergence between the target distribution and the generator ($D_{KL}(p_{Data}||p_G)$), mean (μ_x) and standard deviation (σ_x) of the two distributions as the optimization progresses.

To implement adversarial learning for the VQG instances described above, we followed the `algo2qpu` framework[311], which provides a guideline for the implementation and deployment of quantum algorithms in near-term quantum devices. We started by implementing our variational circuits using the `PyQuil` programming language[313], part of the Forest platform which allows for deployment on both quantum virtual machines (QVM) and quantum processing units (QPUs). The functions for computing expectation values and gradients of the expectation values of variational circuits were encapsulated using the autograd function class available in the `PyTorch` library[253]. This enables integration with the `PyTorch` modules for implementing classical neural networks, applying automatic differentiation and performing gradient-based optimization. In our experiments, we performed simulation of the quantum circuits on the QVM and carried out adversarial learning with the Adam optimizer. We applied typical strategies employed in classical GANs to improve convergence, such as one-sided label smoothing and random flip noise for the discriminator, as described in 7.6(b). In our numerical experiments, the expectations values produced by the generator and quantum discriminator were estimated with 10000 noiseless circuit runs per data point. The real data distribution was generated with expectation values computed up to working precision, as this plays the role of a classical data source.

Figure 7.6(c) illustrates the training of our VQG instance with schemes I (right panel) and II (left panel). We show the dynamics of the discriminator and generator losses during training as well as the Kullback-Leibler (KL) divergence between the generated and the target distributions, computed from the discretized distributions obtained from sampling. We also track the mean and standard deviation during the optimization. For both training schemes, we observe the convergence of the losses to the expected equilibrium point located around $ln(0.5) \approx 0.7$. At the beginning of the training, the learning signal from the discriminator is relatively low and the KL divergence is mostly constant, however, it starts decreasing as the the learning signal rises. We observe that both

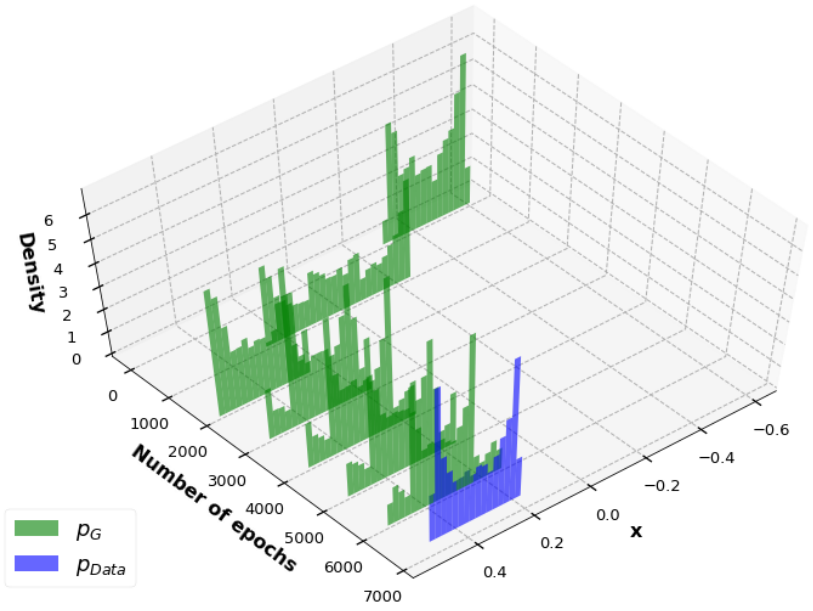


Figure 7.7: Histogram of the data distribution produced by the generator, p_G, at different epochs of the training process (green histograms), compared to the target data distribution, p_{Data}, (blue histogram). We observe how as the optimization progresses, the generated distribution starts resembling the target one. The data corresponds to the optimization with a quantum discriminator (scheme II). All the histograms were computed using the same one thousand samples drawn from $p_z(z) \sim U(-1, 1)$ as noise source.

schemes achieve convergence to an approximation of the target distributions, as evidenced by the evolution of the KL divergence and the distribution moments plotted on Figure 7.6(c). Figure 7.7 compares the distribution produced by the generator with the target distribution, at different moments of the training process for scheme II.

During the simulation, we also tracked the gradients of the the discriminator and generator, noticing that the gradient components in scheme II (quantum discriminator) were around an order of magnitude larger compared to scheme I. Such large gradients can lead to convergence issues during the optimization. In particular, we observe a non-converging oscillatory behavior of the training dynamics in some of the first numerical experiments. This behavior is well documented on the classical GAN literature and is associated to the lack of an incentive for the discriminator to converge once the generator reaches the target distribution[228]. We alleviated this problem by reducing the learning rates for the Adam algorithm, which shifted the dynamics to a damped

oscillation, like the one observed in the right panel of Figure 7.6(c). Standard approaches for treating this problem involves introducing regularization terms for the discriminator cost function on the real data[228,282]

The observation about the magnitude of the gradients can be linked to the difference in the parameterization of variational circuits compared to neural networks. As pointed out in[296], a variational circuits acting on n qubits can be interpreted as a linear unitary layer acting on a vector of size 2^n. Correspondingly, in the language of neural networks, this unitary can be seen as a matrix of complex weights of dimensions $2^n \times 2^n$, parameterized by only $O(poly(n))$ variables, corresponding to the tunable parameters of the variational circuit. In contrast, each of the 4^n entries of the weight matrix is a parameter in a typical layer of a feed-forward neural network. Since gradients are calculated with respect to all the parameters, a learning signal passing through a dense layer of size $2^n \times 2^n$ is distributed among all the 4^n weights. In contrast, the same signal would distribute among only a polynomial number of parameters in the case of the variational circuit, leading to much larger gradient components. This comparison also offers insights into the utility of variational circuits for machine learning as efficient parameterizations of linear transformations of high dimension.

7.4.2 Implementation on NISQ devices

Our proposed VQG model can be implemented on a fault-tolerant quantum computer, but its variational nature makes it especially suitable for implementation on a noisy-intermediate scale quantum devices. As others HQC approaches, the cost of the algorithm is associated to the number of samples required for evaluating and training the model. As pointed out in Section 7.3.1.3, the repetition cost of evaluating the model scales as $O(M/\varepsilon^2)$, where M is the number of operators measured in the decoding step and ε is the precision for each expectation value. A single gradient evaluation scales as $O(n_g M/\varepsilon^2)$, where n_g is the number of parameterized gates in the variational

circuit. For many of the variational circuits discussed here, the number of parameters is linear in the number of qubits, $n_g \in O(N)$. The minimal number of qubits required for the implementation is determined by the number of qubits required by the quantum encoder. For the product encoders used in this work, the number of qubits scales linearly with the size of the noise vector, z. while the depth of the encoding circuit is only constant. One could envision more general variational encoding circuits that trade circuit depth by number of qubits. In the case of amplitude encoding, the number of qubits used is only $O(log(n))$ and $O(n)$ two-qubit gates are required for state preparation.

In addition to the sampling cost, the estimation of the gradients of the variational circuits faces two important challenges: 1) the impact of noise on the estimation of expectation values and 2) the recent observation that the gradients of near-random variational circuits tend to vanish with a probability exponential on the number of qubits, an phenomenon known as *barren plateau* of the quantum neural network training landscapes[223]. To address the issue of noise in the VQG implementation, we could apply some of the recent proposals for error mitigation in the estimation of expectation values on NISQ devices. The basic principle of these proposals is that the first order contributions of the noise to the expectation values can be removed by introducing a controllable source of noise in the circuit of interest[98,331]. The expectation values are estimated at different error levels and an extrapolation to zero noise is performed using simple regression techniques. These methods have been already applied in experiments for VQE and variational quantum classification[169]. While error mitigation could benefit the gradient estimation, we also point out that noise is generally included in practice to improve convergence during the GAN training[228]. Most likely, the training process will be able to tolerate moderate levels of noise, as observed in the case of variational circuits and tensor networks employed in classifications tasks[123,153,296]. Error mitigation will likely play a more crucial to generate high quality samples after training is

complete.

In our proof of principle experiments, we did not observe vanishing gradients most likely due to the small size of the circuits used in the example. In larger scale implementation of the VQG model, barren plateaus might become an issue. In this case, several strategies could be employed to mitigate the problem. One of them, especially suitable for variational circuits built on circuit blocks, is *block-by-block* training. In this case, the optimization starts with a variational circuit with a single or a few circuit blocks, which are less likely to suffer from the barren plateau issue due to the relatively small number of parameters. In subsequent rounds, we add more blocks to the variational circuit and use the optimal variational parameters of the previous round to initialize the new round of training. This procedure can improve convergence, as shown in the case of classical deep neural networks[33,139].

Another strategy is to use circuits with subcomponents that admit classical simulation or inspired by classically simulable circuits. An example of such circuits is the low-depth circuit ansatz (LDCA) proposed in Ref.[78] for quantum simulation of fermions. The basic building block of the circuit is composed of matchgates[339], that can be simulated classically, augmented with a set *ZZ* rotations that increase the complexity of the circuit. Therefore, we could run classical simulations of the VQG training with an LDCA variational circuits without the *ZZ* interactions. The optimal parameters obtained from the classical simulation can be then employed to initialize the training with the full LDCA circuit using the quantum computer. A similar procedure can be applied to variational circuits based on tensor networks, that admit simulation on classical computers with small bond dimensions, as suggested in Refs.[123,153].

7.5 Conclusion and outlook

In this chapter we have presented a hybrid quantum-classical architecture, comprising variational quantum circuits and classical functions, for modeling continuous probability distributions. Our variational quantum generator incorporates two quantum circuits: a quantum circuit encoding a classical random variable into a quantum state, $R(z)$, and a variational circuit $G(\Theta_g)$, whose parameters are optimized to mimic the target classical probability distribution. A sample, x_{Fake} from the VQG architecture is generated sampling z from a noise distribution $z \sim p_z(z)$, encoding this variable into a quantum state using the encoding circuit, applying the variational circuit and measuring the expectation values of a set of predefined operators. The vector of expectation values, P, can be post-processed using a classical function, $f_g(P)$, such as a neural network, to generate the sample $f_g(P) = x_{Fake}$. The VQG architecture can be trained using a gradient-based adversarial learning strategy, where a second model, known as discriminator, compares the quality of the samples generated by the VQG model with samples from the real data distribution. We show that the required gradients can be calculated using existing techniques for evaluating gradients of variational functions and exploiting the established infrastructure for automatic differentiation of classical functions. We illustrate this process with a simple proof-of-principle experiment where a VQG instance with fixed parameters serves as the target distribution.

Our proposal contributes to an increasing body of work exploring the use of hybrid quantum-classical computing in machine learning, offering an approach to perform generative modeling of continuous probability distributions with quantum computers. Furthermore, the same architecture employed in VQG can be used to build models for classification, as illustrated in Section 7.3.2. We also present a strategy for training our proposed generator with both classical and quantum discriminators, taking advantage of the integration of gradient estimation of variational circuits and automatic differentiation strategies. The incorporation of these tools can

also benefit the implementation of other HQC algorithms, such as VQE. Nowadays, the availability of libraries for programming and executing quantum circuits[69,270,313,322] that are compatible with standard libraries for machine learning[2,253], facilitate this integration. Recently, specialized libraries for automatic differentiation of variational circuits has been also developed[35].

There are several open questions that remain to be investigated. Perhaps the most significant one is whether this approach can offer an advantage with respect to purely classical models for generative learning. Some theoretical works[43,95] indicate that variational circuits might bear an advantage for discrete generative tasks, however the extent to which this impacts practical applications such as image, sound and language generation will require extensive computational studies on real instances. A second aspect that needs to be studied is the role of noise the performance of the VQG model implemented on NISQ devices. As discussed in Section 7.4.2, one of the strategies to improve convergence during classical GAN training is to introduce noise in the data, which generally prevents over-fitting in the discriminator and improve the robustness of the final model. The extent to which noise on NISQ devices can be tolerated or can benefit the training process, as well as the overall quality of the the distributions generated by VQG, most likely depends on the specific noise process and the nature of the target distribution. Finally, a third research direction is the study of adversarial learning of classical generators using quantum discriminators, in particular how this particular arrangement could impact the convergence of the training process.

An important advantage of the VQG model is its flexibility, allowing for exploring multiple choices of variational circuits and encodings. This also allows for designing new strategies to incorporate non-linearities through hybrid quantum-classical architectures. The architecture presented in this work incorporates non-linearities through pre-processing classical data with non-linear functions and through the tensorial mapping approach. The variational circuit acts as a

linear layer but non-linearities can be incorporated after measurement via classical post-processing. Both, variational circuits and classical post-processing can be considered together as a single non-linear layer and can be repeated to build deep hybrid quantum-classical architectures. Another important future research direction is the design of new circuits for variational encodings that can balance the cost of number of qubits and depth while incorporating non-linearities. In the specific case of VQG, the quantum encoder, $R(z)$, plays the role of noise source, sampling randomly a state from the manifold defined by the encoding circuit. An alternative strategy could be replacing the encoding circuit with an efficient circuit to approximately sample from the Haar measure, for example, an efficiently implementable unitary 2-design[82].

Finally, the most thrilling aspect of the VQG approach is the prospect of realizations of the algorithm on quantum devices to tackle standard problems in generative learning. As NISQ devices approach sizes that surpass the possibility of classical simulation, quantum algorithms that allow for gradually incorporating quantum capabilities into the established quantum machine learning pipelines will be required. The VQG model offers such a framework, establishing a strategy to combine increasingly large variational circuits with standard neural networks to model data distributions. Future work will be dedicated to exploring the utility of the VQG approach in specific scientific and industrial applications, including image processing, finance, medicine, cybersecurity, and drug and materials design.

7.6 Appendix

7.6.1 Variational circuits architectures

Figure 7.8 describes some examples of variational circuits employed in HQC algorithms. Figure 7.8(a) shows a circuit block containing all the possible controlled one-qubit rotations among a set of qubits, interleaved with a set of single qubit rotations. We start considering the rotations

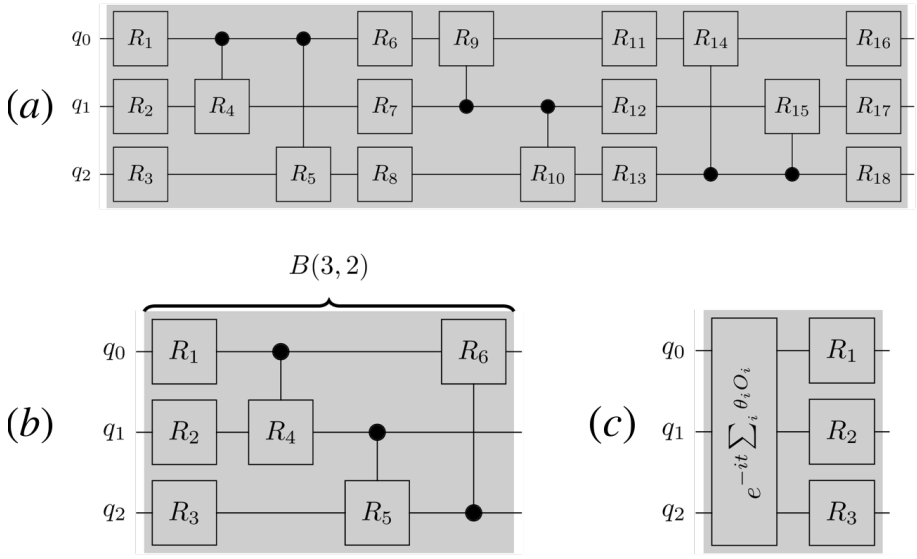


Figure 7.8: Examples of variational circuit blocks employed in quantum machine learning: a) Circuit block employed for HQC algorithms such as QAE and QVECTOR[163,280]. b) Generalization of circuit blocks proposed by Schuld et al.[296]. The depicted blocks corresponds to 3 qubits with range 2 ($B(3,2)$). c) Variational circuit corresponding to quantum evolution under a Hamiltonian with tunable parameters (θ_j) and single-qubit rotations. R_j represents a generic single-qubit gate.

controlled by the first qubit, followed by the rotations controlled by the second qubit and so on. The number of parameters in this circuit block scales as $O(n^2)$. More simplified circuit blocks, where entangling operations are not parameterized, has been used for experimental demonstration of QML for classification[138]. In this case, the disposition of the entangling gates is generally dictated by the constraints in the qubit connectivity of the processor.

Families of circuit blocks have been also proposed. In particular, Schuld et al. proposed a series of circuit blocks for classification, generically referred to as code blocks $B(n,r)$g[296]. An example of these blocks is depicted in 7.8(b). A block $B(n,r)$ comprises a layer of general single-qubit rotations $R = R(\alpha,\beta,\gamma)$ applied to each of the n qubits of the register followed by a layer of $n/\gcd(n,r)$ controlled-R gates, where r is the range of the two-qubit gates and $\gcd(n,r)$ is the greatest common divisor of n and r. The target and control qubits of the j-th two-qubit gate in the block are given by the integers $t_j = (jr-r) \mod n$ and $c_j = jr \mod n$, respectively. This

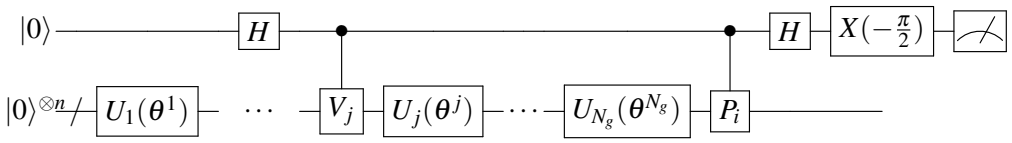


Figure 7.9: Circuit for measuring $\frac{\partial\langle 0|U(\Theta)^\dagger P_i U(\Theta)|0\rangle}{\partial\theta^j}$. The gate $X(\frac{\pi}{2})$ rotates to the Y-basis.

definition guarantees a number of parameters that scales linearly with the size of the qubit register, n. In general, B-blocks are capable of entangling/unentangling all the qubits with numbers that are multiple of $\gcd(n,r)$. If n and r are co-prime, the network of entangling gates forms a cycle graph capable of entangling/unentangling all the qubits in the register. Finally, 7.8(c) presents and example of a generic circuit implementing evolution under a local Hamiltonian, where the coefficient of the Hamiltonian terms serve as variational parameters.

7.6.2 Estimation of analytical gradients for variational circuits

Recently, it has been shown that gradients of expectation values of variational circuits can be estimated analytically using slight modifications of the initial quantum circuit. Specifically, consider a variational circuit $U(\Theta)$ of the form:

$$U(\Theta) = U_1(\theta^1)U_2(\theta^2)\cdots U_{N_g}(\theta^{|\Theta|}), \tag{7.18}$$

where $U_j(\theta^j) = \exp(-i\theta^j V_j/2)$ with $\{V_j\}$ being a Pauli operator. One strategy for computing gradients with respect to the parameter θ^i is using the circuit of Figure 7.9, which requires one additional qubit compared to the original variational circuit[279,296]. Taking measurements on this ancilla qubit provides an estimate to one element of the Jacobian of the vector P, $\nabla P_{i,j}$.

Alternatively, the same component can be estimated using two separate evaluations of the expectation values where the original variational circuit is replaced by the modified circuits

${}^{+}U_j(\Theta)$ and ${}^{-}U_j(\Theta)$[231] defined as:

$$ {}^{+}U_j(\Theta) = U(\theta^1)\cdots U_j\left(\theta^j + \frac{\pi}{2}\right)\cdots U(\theta^{|\Theta|}), \tag{7.19} $$

$$ {}^{-}U_j(\Theta) = U(\theta^1)\cdots U_j\left(\theta^j - \frac{\pi}{2}\right)\cdots U(\theta^{|\Theta|}). \tag{7.20} $$

such as:

$$ \frac{\partial \langle U(\Theta)^\dagger |P_i| U(\Theta)\rangle}{\partial \theta^j} = $$

$$ \mathrm{Re}\left(\langle {}^{+}U_j(\Theta)^\dagger |P_i| {}^{+}U_j(\Theta)\rangle - \langle {}^{-}U_j(\Theta)^\dagger |P_i| {}^{-}U_j(\Theta)\rangle\right) \tag{7.21} $$

This approach has been recently coined as a classical linear combination of unitaries (CLCU)[295]. Compared to the circuit of Figure 7.9, the CLCU strategy requires twice as many measurements to estimate the gradient to the same accuracy. However, it does not require an ancilla qubit and employs the same circuit as the objective function, $U(\Theta)$. The later implies that the compiled of ${}^{+}U(\Theta)$ and ${}^{-}U(\Theta)$ are the same as $U(\Theta)$, which might simplify the implementation.

7.6.3 Automatic differentiation

Automatic differentiation (AD)[95,240] is an algorithmic strategy to extend a program that computes numerical values of a function such as it can also compute arbitrary derivatives of the same function. Unlike numerical differentiation, AD provides exact derivatives up to working computational precision. AD also differs from symbolic differentiation in the sense that it computes numerical values of the derivatives instead of analytical expressions. To achieve its goal, AD extends the domain of variables in the computation to incorporate derivative values and introduces a programming semantics to enable the propagation of the derivatives using the chain rule. This allows to compute arbitrary derivatives of any function by applying differentiation to the

sequence of elementary operations and elementary functions that implement the function on the computer. The process is performed automatically during execution time and has only a constant overhead in computational cost compared to the execution of the original function.

There exist different strategies for implementing AD. For the work presented in this chapter, we implemented a strategy known as reverse mode accumulation[95], which is a generalization of the back-propagation procedure employed in feed-forward neural networks[120]. In this case, the calculation of a function is broken down into a series of intermediate steps with results stored by intermediate variables. The inter-dependence of these variables constitutes a directed graph known as computational graph. In reverse mode, the function and its derivatives are calculated in two steps known as the *forward* and *backward* passes. In the forward pass, the original function code is run, computing the values of all the intermediate variables and recording the dependencies of the computational graph using a book-keeping procedure. In the backward pass, derivatives are calculated by computing the derivatives of each intermediate variable with respect to its immediate inputs and propagating the derivatives in reverse through the computational graph, from outputs to inputs.

Part III

Outlook and conclusions

This page intentionally left blank.

8

Future research directions

As expressed in the introduction to this dissertation, the field of variational quantum algorithms is still in his infancy and is currently mainly driven by the need to find useful applications for NISQ devices. Achieving useful quantum computing in the near-term is vital to secure the resources and maintain the momentum necessary for building full-error corrected quantum computers, as indicated by the most recent report in the progress of quantum information technologies by the National Academy of Sciences[129]. Along the path towards useful quantum computing with variational algorithms, many fundamental scientific and engineering problems need to be solved. Here we describe what we believe are some of the main challenges in variational quantum algorithms as well as research directions that might help to overcome them, having as starting point the research presented in the primary research chapters. We have grouped these directions into two categories: 1) research directions that impact the entire family of variational quantum algorithms and 2) research directions for applications of variational quantum algorithms.

In the first category, we include ideas that we consider to a specific extend independent of a particular application and pertain to the general scheme of variational algorithms described in Section 1.2. In the second category, we include specific research directions for variational algorithms in a particular niche of applications, such as quantum simulation or machine learning. While this separation is somewhat arbitrary, it highlights that variational algorithms, independent of their application, follow approximately the same blueprint and therefore identifying

improvements to that blueprint can bring benefits to the entire family of variational algorithms. Naturally, this way of thinking does not exclude the possibility that certain improvements developed initially for a particular application niche could be generalized or even directly applied in other niches.

Future research directions in quantum variational algorithms

As described in Section 1.2, variational algorithms incorporate three elements: 1) a variational circuit, 2) functional estimation and 3) an optimization loop. We identify research directions to improve in these three fronts:

Variational circuits: most variational circuit ansatzes were originally conceived for a particular functional, for example UCC for VQE in Chemistry (Chapter 2). An important question around ansatz design is if there are general properties that define a "good" variational ansatz somewhat independent of the application. For example, given that entanglement is a resource for quantum algorithms, one might want a variational circuit to generate entanglement effectively using the set of gates that better suits the experimental constraints of a particular platform. The entangled states generated are likely to have certain structure or reflect specific symmetries, as opposed to being completely random. There is also a certain notion of variational freedom in circuits, meaning how much of the Hilbert space can be explored by a specific circuit parameterization, as well as how sensible the circuit is to changes in the parameters. Properties like these might determine the effectiveness of the circuit and influence our ability to optimize it. Defining useful metrics to qualify variational circuits independently of the functional is an important line of research in variational quantum computing.

The problem of designing variational circuits can also be tackled from a more practical perspective using adaptive techniques to generate variational circuits on the fly. We could achieve

this, for instance, by adopting hyperparameter optimization techniques from classical machine learning. In this case, hyperparameters are parameters of the model fixed before starting the optimization, for example, the number of layers and their side in a classical neural network. In variational circuits, the layout of the gates and set of gates composing a circuit can play this role. By using classical methods such as Bayesian optimization[36,155,314] and evolutionary strategies[157,230], we could design variational circuits layouts given a set of gates and a particular connectivity. In practice, we could obtain a general layout of the circuit by benchmarking small problem instances on small quantum computers or through classical simulators. The designs obtained in this first stage can be the starting point for subsequent refinements obtained from computations on larger devices.

Functional estimation: Although the particular application of the algorithm defines the form of the functional, functional estimation involves general aspects that are common to all the variational approaches. Ultimately, the evaluation of any functional reduces to collecting measurements from the device. For many applications, the goal is to measure observables up to some accuracy, which in cases like VQE is very costly, as described in Chapter 2. Finding methods that make the most out of a limited number of noisy measurements can improve general performance. One direction we started to explore in Chapter 4 was the use of classical statistical models to make inference on the data obtained from the device; specifically, we employed regression techniques to remove systematic biases and partially reconstruct the functional landscape to help optimization. In that spirit, exploring the potential application of classical statistical inference tools in the post-processing step of variational algorithms constitutes a promissory research direction.

An already ongoing research direction for functional estimation is error mitigation. These techniques aim at correcting first order errors in observables estimated on NISQ devices employing regression techniques applied to data collected at different noise levels[98,169,331]. Finding the most

effective way to integrate these techniques into the general workflow of variational algorithms is a natural next step. A topic related to error mitigation is the observation that variational algorithms can be robust to specific errors in the device. While this is intuitive in the case of calibration errors in gates, whether this is true for other types of errors and the mechanism behind such corrections are research questions that might lead to insights for designing variational circuits that exhibit such robustness.

Optimization techniques: As mentioned in the introduction, the field of non-linear optimization methods is vast[246]. Since optimization is a critical aspect of variational algorithms, investigating both theoretically and numerically the performance of a variety of optimization methods for variational algorithms is of vital importance. Optimizers that can handle noisy data are particularly important. Along the same lines, the incorporation of techniques for reducing the cost of estimating functionals and their derivatives is critical. Methods such as stochastic gradient descent could help in the case of functionals expressed as weighted averages of many observables, which is the case of Hamiltonians in VQE, for example. Finally, a central problem in the development of optimization techniques for variational algorithms is the existence of barren plateaus in the functional landscape of variational circuits[223]. While we have mentioned some strategies that might help to fight this problem in Chapter 7, this will continue being an issue especially as we scale variational algorithms to bigger quantum devices. Finding effective methods to combat the vanishing gradient problem in variational circuits is of vital importance for the field.

Future directions in applications for variational algorithms

We identify four important application areas for variational algorithms: quantum simulation, quantum optimization, machine learning, and quantum computing. We briefly describe the current state of these application areas along with some tentative directions.

Quantum simulation: Research in this area has focused on the development of the VQE algorithm. Some of the recent advances in this matter include the development of new ansatzes[19,78,179], extensions of the algorithm to study quantum dynamics using some formulations of the time-dependent variational principle[368], the development of different schemes for error mitigation[98,169,221,331], extensions of the algorithm to compute excited states[70,99,146,224,283,287], and integration with existing techniques in quantum chemistry to tackle larger systems[366]. Most of these advances have been tested using numerical simulations. Implementing these proposals on actual quantum processors is a non-straightforward next step necessary to assess the practical value of these proposals. Another important research direction concerns the calculation of molecular properties using VQE, for example, the study of thermochemistry and kinetics. In the same direction, the software infrastructure to scale up VQE calculations and interface them with existing simulations tools in Chemistry will require further development as well. Finally, the application of VQE to problems in condensed matter physics, such as superconductivity, is another promissory area of research that is underdeveloped compared to Quantum Chemistry.

Quantum Optimization: Here, most of the recent advances have focused on developments of QAOA. Recently, there have been extensive studies on the applicability of QAOA to Max-Cut problems and its relation to quantum annealing, indicating that the optimal parameters for QAOA tend to concentrate around specific values[46,373] and showing a separation between QAOA and quantum annealing[75,211,326]. The variational circuit employed in QAOA has been applied to specific problems in machine learning[354,363] and more recently, to solve the factoring problem, leading to a new algorithm called Variational Quantum Factoring (VQF)[9], able to run on NISQ devices. In this direction, there is significant room for exploring the application of QAOA to other constraint optimization problems.

Quantum machine learning: Recently, the field of quantum machine learning with variational

circuits have experienced considerable growth. Most of the work in this direction has focused on using variational circuits for classification tasks[298]. As shown in Chapter 7, a few proposals in generative algorithms using variational circuits have also appeared, including VQG. From this body of work, it is clear that variational algorithms can solve a wide range of tasks in machine learning. Whether these variational approaches can offer a practical or proven asymptotic advantage is a matter of ongoing research. From a practical point of view, an essential direction towards this goal is the application of these approaches to established machine learning benchmarks datasets. While it would be hard to test these approaches on real quantum devices, a way to systematically perform such tests is by the use of simulators and the study of hybrid architectures, such as VQG, that interface a small quantum computer with a large neural network. In the theory front, recent work has shown complexity separations between variational circuits and neural networks in generative tasks[95]. An important research direction would be to design simple machine learning models where the power of variational algorithms can be studied analytically. These small toy problems might not be practical but could lead to a better understanding of the learnability and expressiveness of variational circuits compared to classical machine learning models.

Quantum computing: Despite being a natural niche for applications pertaining operations on quantum computers (that by definition can be executed naturally on quantum devices) variational algorithms in this domain reduce to techniques for engineering quantum gates[145,201] and to study quantum foundations[13], in addition to QVECTOR and QAE. A significant challenge for the assessment of these approaches is the need for larger quantum computers, particularly in the case of QVECTOR. In this case, the goal is to train a model that learns from the particular noise processes happening on the device. While we get to the point of testing these techniques in sufficiently large quantum computers, the use of classical high-performance simulators

implementing very general noise models might offer a first testbed to develop working implementations of the algorithms, which can be subsequently deployed on medium to sizeable real quantum processors when these become available.

Other applications: variances of the VQE algorithms have been proposed to address computational problems such as matrix-matrix multiplication [100] and diagonalization of quantum states [193]. Developing variational quantum algorithms for new areas of application is a constant driver for the field.

As a final remark, we point out the importance that all the theoretical and practical developments in variational algorithms come with implementations on robust software frameworks. The simultaneous development of theory and software allows for quick benchmarking and deployment of these techniques on quantum hardware, ultimately accelerating progress towards the practical frontier.

9

Conclusions

Quantum computing is a computational model that employ quantum systems to encode and process information. The unique properties of these systems allow for computational speed-ups for many significant problems, including quantum simulation, factoring, among others. The prospect of solving these problems and the development of techniques to control quantum systems motivated the experimental realization of quantum computers. Nowadays, these machines are improving quickly. However, building fault-tolerant quantum computers require a significant number of resources, establishing this goal as a still long term promise. In the process of reaching this goal, we are building quantum devices that will soon surpass the limit of simulability with classical supercomputers. How do we harness the computational power of these NISQ devices? The answer might lie in variational quantum algorithms, a new paradigm in quantum algorithm design. These approaches aim at solving computational problems that encode their solutions as extreme values of a functional. By representing functions as parameterized quantum circuits and finding a set of parameters that optimize the functional value, variational algorithms find approximate solutions to these problems. The flexibility in the choice of heuristic for the variational circuit allows to expands the operability of these approaches within the regime of the coherence times of NISQ devices, avoiding the necessity of error-correction.

In this dissertation, we presented a general formulation of variational algorithms and provided examples of the application of this paradigm to tackle problems in multiple fields, including

quantum simulation, quantum computing, and statistical modeling. In the first part, we study the application of the variational quantum eigensolver, the first variational quantum algorithm to be proposed[255], to the calculation of the ground state of molecular systems. We investigated the implementation of VQE using a method called unitary coupled cluster to build the variational circuit. We exploited existing techniques in the area of quantum chemistry to improve the cost of implementing this approach and developed a method for estimating gradients of variational circuits to aim optimization. We also demonstrated some of these techniques on real quantum processors, realizing some of the first scalable quantum computations of molecular energies. Finally, we improved the prospects of achieving VQE calculation for larger systems, both in chemistry and condensed matter physics, by proposing a new variational ansatz with more modest resource requirements that can describe strongly correlated fermionic systems.

In the second part, we expanded the potential applications of the variational algorithm paradigm to problems in quantum computing and statistical modeling. Inspired by the autoencoder model of machine learning, we develop a new variational algorithm called the quantum autoencoder (QAE). This method is capable of learning quantum circuits to compress an ensemble of quantum states efficiently. When applied to problems in quantum simulation, QAE provides an unsupervised technique to design more efficient state preparation circuits. Drawing similar inspiration from machine learning, we created another quantum algorithm for finding quantum encoding and recovery circuits for error-correction, called the variational quantum error corrector (QVECTOR). By training variational circuits as encoding and recovery maps based on the behavior of the actual quantum device, QVECTOR provides mappings tailored to the needs of a particular quantum processor, opening a path to potentially more efficient and effective protocols to protect quantum computers from errors.

Finally, we explored the applications of the variational quantum computing paradigm to

problems in classical machine learning. Specifically, we proposed a quantum model called variational quantum generator (VQG), that combines a variational circuit with a classical neural network to model classical probability distributions. We showed that VQG could be trained using an adversarial learning approach, where another neural network or variational circuit judge the quality of the samples generated by the model. Our method is a blueprint for creating hybrid quantum-classical models in machine learning, opening a new practical path to study the potential of quantum computers to solve classical machine learning problems.

The research presented in this work provides insights into the challenges of developing and implementing variational quantum algorithms for near-term quantum computers and offers examples of strategies to address some of these challenges. We hope that our work inspires other researches to explore the path of variational quantum algorithms, providing some practical tools to help them advance the algorithms studied here or propose their own. We believe that by leveraging the variational approach to quantum computing, we will be able to push this field towards the practical frontier.

Bibliography

[1] S. Aaronson and L. Chen. Complexity-theoretic foundations of quantum supremacy experiments. arXiv preprint arXiv:1612.05903 (2016).

[2] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: a system for large-scale machine learning. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation*, volume 16, pages 265–283, 2016.

[3] A. Abeyesinghe, I. Devetak, P. Hayden, and A. Winter. The mother of all protocols: restructuring quantum information's family tree. *Proc. R. Soc. A*, **465**, 2537 (2009).

[4] D. S. Abrams and S. Lloyd. Simulation of many-body fermi systems on a universal quantum computer. *Phys. Rev. Lett.*, **79**, 2586 (1997).

[5] M. L. Abrams and C. D. Sherrill. Natural orbitals as substitutes for optimized orbitals in complete active space wavefunctions. *Chem. Phys. Lett.*, **395**, 227 (2004).

[6] D. Aharonov and M. Ben-Or. Fault-tolerant quantum computation with constant error rate. *SIAM J. Comput.*, **38**, 1207 (2008).

[7] P. W. Anderson. The resonating valence bond state in la_2cuo_4 and superconductivity. *Science*, **235**, 1196 (1987).

[8] P. W. Anderson, P. A. Lee, M. Randeria, T. M. Rice, N. Trivedi, and F. C. Zhang. The physics behind high-temperature superconducting cuprates: The plain "plain vanilla" version of rvb. *J. Phys.: Condens. Matter*, **16**, 755 (2004).

[9] E. R. Anschuetz, J. P. Olson, A. Aspuru-Guzik, and Y. Cao. Variational quantum factoring. arXiv preprint arXiv:1808.08927 (2018).

[10] B. R. Arnold and J. Michl. Ultraviolet and polarized infrared spectroscopy of matrix-isolated cyclobutadiene and its isotopomers. *J. Phys. Chem.*, **97**, 13348 (1993).

[11] B. R. Arnold and J. Michl. *Kinetics and spectroscopy of carbenes and biradicals*. Springer Us, 2013.

[12] B. R. Arnold, J. G. Radziszewski, A. Campion, S. S. Perry, and J. Michl. The raman spectrum of matrix-isolated cyclobutadiene. evidence for environmental hindrance to heavy-atom tunneling? *J. Am. Chem. Soc.*, **113**, 692 (1991).

[13] A. Arrasmith, L. Cincio, A. T. Sornborger, W. H. Zurek, and P. J. Coles. Variational consistent histories: A hybrid algorithm for quantum foundations. arXiv preprint arXiv:1812.10759 (2018).

[14] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon. Simulated quantum computation of molecular energies. *Science*, **309**, 1704 (2005).

[15] A. Aspuru-Guzik, R. Lindh, and M. Reiher. The matter simulation (r)evolution. *ACS Central Science*, **4**, 144 (2018).

[16] R. Babbush, D. W. Berry, I. D. Kivlichan, A. Y. Wei, P. J. Love, and A. Aspuru-Guzik. Exponentially more precise quantum simulation of fermions in second quantization. *New J. Phys.*, **18**, 033032 (2016).

[17] R. Babbush, D. W. Berry, Y. R. Sanders, I. D. Kivlichan, A. Scherer, A. Y. Wei, P. J. Love, and A. Aspuru-Guzik. Exponentially more precise quantum simulation of fermions in the configuration interaction representation. *Quantum Sci. Technol.*, **3**, 015006 (2018).

[18] R. Babbush, J. McClean, D. Wecker, A. Aspuru-Guzik, and N. Wiebe. Chemical basis of trotter-suzuki errors in quantum chemistry simulation. *Phys. Rev. A*, **91**, 022311 (2015).

[19] R. Babbush, N. Wiebe, J. McClean, J. McClain, H. Neven, and G. K.-L. Chan. Low-depth quantum simulation of materials. *Phys. Rev. X*, **8**, 011044 (2018).

[20] J. Bang, J. Ryu, S. Yoo, M. Pawłowski, and J. Lee. A strategy for quantum algorithm design assisted by machine learning. *New. J. Phys*, **16**, 073017 (2014).

[21] J. Bardeen, L. N. Cooper, and J. R. Schrieffer. Microscopic theory of superconductivity. *Phys. Rev.*, **106**, 162 (1957).

[22] R. Barends, J. Kelly, A. Megrant, D. Sank, E. Jeffrey, Y. Chen, Y. Yin, B. Chiaro, J. Mutus, C. Neill, P. O'Malley, P. Roushan, J. Wenner, T. C. White, A. N. Cleland, and J. M. Martinis. Coherent josephson qubit suitable for scalable quantum integrated circuits. *Phys. Rev. Lett.*, **111**, 080502 (2013).

[23] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, Y. Campbell Chen, Z. Chen, B. Chiaro, A. Dunsworth, C. Neill, P. O'Malley, P. Roushan, A. Vainsencher, J. Wenner, A. N. Korotkov, A. N. Cleland, and J. Martinis. Superconducting quantum circuits at the surface code threshold for fault tolerance. *Nature*, **508**, 500 (2014).

[24] R. Barends, L. Lamata, J. Kelly, L. García-Álvarez, A. G. Fowler, A. Megrant, E. Jeffrey, T. C. White, D. Sank, J. Y. Mutus, et al. Digital quantum simulation of fermionic models with a superconducting circuit. *Nat. Commun.*, **6**, 7654 (2015).

[25] R. J. Bartlett, S. A. Kucharski, and J. Noga. Alternative coupled-cluster ansätze ii. the unitary coupled-cluster method. *Chem. Phys. Lett.*, **155**, 133 (1989).

[26] R. J. Bartlett and M. Musial. Coupled-cluster theory in quantum chemistry. *Rev. Mod. Phys.*, **79**, 291 (2007).

[27] B. Bauer, D. Wecker, A. J. Millis, M. B. Hastings, and M. Troyer. Hybrid quantum-classical approach to correlated materials. *Phys. Rev. X*, **6**, 031045 (2016).

[28] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: a survey. *J. Mach. Learn. Res.*, **18**, 1 (2018).

[29] A. T. Bell. Challenges for the application of quantum chemical calculations to problems in catalysis. *Mol. Phys.*, **102**, 319 (2004).

[30] A. Ben-Aroya and A. Ta-Shma. Quantum expanders and the quantum entropy difference problem. arXiv preprint quant-ph/0702129 (2007).

[31] M. Bender, P.-H. Heenen, and P.-G. Reinhard. Self-consistent mean-field models for nuclear structure. *Rev. Mod. Phys.*, **75**, 121 (2003).

[32] M. Benedetti, D. Garcia-Pintos, Y. Nam, and A. Perdomo-Ortiz. A generative modeling approach for benchmarking and training shallow quantum circuits. arXiv preprint arXiv:1801.07686v2 (2018).

[33] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems 20*, pages 153–160, 2007.

[34] J. Benhelm, G. Kirchmair, C. F. Roos, and R. Blatt. Towards fault-tolerant quantum computing with trapped ions. *Nat. Phys.*, **4**, 463 (2008).

[35] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, and N. Killoran. Pennylane: Automatic differentiation of hybrid quantum-classical computations. arXiv preprint arXiv:1811.04968 (2018).

[36] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems 24*, pages 2546–2554, 2011.

[37] A. Bermudez, X. Xu, R. Nigmatullin, J. O'Gorman, V. Negnevitsky, P. Schindler, T. Monz, U. G. Poschinger, C. Hempel, J. Home, F. Schmidt-Kaler, M. Biercuk, R. Blatt, S. Benjamin, and M. Müller. Assessing the progress of trapped-ion processors towards fault-tolerant quantum computation. *Phys. Rev. X*, **7**, 041061 (2017).

[38] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd. Quantum machine learning. *Nature*, **549**, 195 (2017).

[39] C. M. Bishop. *Pattern recognition and machine learning*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[40] R. Blatt and C. Roos. Quantum simulations with trapped ions. *Nat. Phys.*, **8**, 277 (2012).

[41] B. B. Blinov, D. Leibfried, C. Monroe, and D. J. Wineland. Quantum computing with trapped ion hyperfine qubits. *Quantum Inf. Process.*, **3**, 45 (2004).

[42] C. Bloch and A. Messiah. The canonical form of an antisymmetric tensor and its application to the theory of superconductivity. *Nuclear Physics*, **39**, 95 (1962).

[43] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven. Characterizing quantum supremacy in near-term devices. *Nat. Phys.*, **14**, 595 (2018).

[44] H. Bombín. Gauge color codes: optimal transversal gates and gauge fixing in topological stabilizer codes. *New J. Phys.*, **17**, 083002 (2015).

[45] H. Bombin and M. A. Martin-Delgado. Topological quantum distillation. *Phys. Rev. Lett.*, **97**, 180501 (2006).

[46] F. G. Brandao, M. Broughton, E. Farhi, S. Gutmann, and H. Neven. For fixed control parameters the quantum approximate optimization algorithm's objective function value concentrates for typical instances. arXiv preprint arXiv:1812.04170 (2018).

[47] S. Bravyi, J. M. Gambetta, A. Mezzacapo, and K. Temme. Tapering off qubits to simulate fermionic hamiltonians. arXiv preprint arXiv:1701.08213 (2017).

[48] S. B. Bravyi and A. Y. Kitaev. Fermionic quantum computation. *Ann. Phys.*, **298**, 210 (2002).

[49] D. J. Brod. Efficient classical simulation of matchgate circuits with generalized inputs and measurements. *Phys. Rev. A*, **93**, 062332 (2016).

[50] B. J. Brown, N. H. Nickerson, and D. E. Browne. Fault-tolerant error correction with the gauge color code. *Nat. Commun.*, **7**, 12302 (2016).

[51] K. R. Brown, J. Kim, and C. Monroe. Co-designing a scalable quantum computer with trapped atomic ions. *npj Quantum Inf.*, **2**, 16034 (2016).

[52] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf. Quantum fingerprinting. *Phys. Rev. Lett.*, **87**, 167902 (2001).

[53] K. Burke and L. O. Wagner. Dft in a nutshell. *Int. J. Quantum Chem.*, **113**, 96 (2013).

[54] J. Bylander, S. Gustavsson, F. Yan, F. Yoshihara, K. Harrabi, G. Fitch, D. G. Cory, Y. Nakamura, J.-S. Tsai, and W. D. Oliver. Noise spectroscopy through dynamical decoupling with a superconducting flux qubit. *Nat. Phys.*, **7**, 565 (2011).

[55] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, **16**, 1190 (1995).

[56] T. Byrnes and Y. Yamamoto. Simulating lattice gauge theories on a quantum computer. *Phys. Rev. A*, **73**, 022328 (2006).

[57] C. Cafaro and P. van Loock. Approximate quantum error correction for generalized amplitude-damping errors. *Phys. Rev. A*, **89**, 022316 (2014).

[58] A. R. Calderbank and P. W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, **54**, 1098 (1996).

[59] L. Cao. Data science: a comprehensive overview. *ACM Computing Surveys (CSUR)*, **50**, 43 (2017).

[60] Y. Cao, G. G. Guerreschi, and A. Aspuru-Guzik. Quantum neuron: an elementary building block for machine learning on quantum computers. arXiv preprint arXiv:1711.11240 (2017).

[61] Y. Cao, J. Romero, and A. Aspuru-Guzik. Potential of quantum computing for drug discovery. *IBM J. Res. Dev.*, **PP**, 1 (2018).

[62] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya, et al. Quantum chemistry in the age of quantum computing. arXiv preprint arXiv:1812.09976 (2018).

[63] J. Casanova, A. Mezzacapo, L. Lamata, and E. Solano. Quantum simulation of interacting fermion lattice models in trapped ions. *Phys. Rev. Lett.*, **108**, 190502 (2012).

[64] G. K.-L. Chan. Low entanglement wavefunctions. *Wiley Interdiscip. Rev. Comput. Mol. Sci.*, **2**, 907 (2012).

[65] H. Chen, L. Wossnig, S. Severini, H. Neven, and M. Mohseni. Universal discriminative quantum neural networks. arXiv preprint arXiv:1805.08654 (2018).

[66] I. L. Chuang and Y. Yamamoto. Quantum bit regeneration. *Phys. Rev. Lett.*, **76**, 4281 (1996).

[67] A. Church. An unsolvable problem of elementary number theory. *Am. J. Math.*, **58**, 345 (1936).

[68] J. I. Cirac, T. Pellizzari, and P. Zoller. Enforcing coherent evolution in dissipative quantum dynamics. *Science*, **273**, 1207 (1996).

[69] Cirq. `https://github.com/quantumlib/Cirq`.

[70] J. Colless, V. Ramasesh, D. Dahlen, M. Blok, J. McClean, J. Carter, W. A. de Jong, and I. Siddiqi. Implementing a variational quantum eigensolver using superconducting qubits. In *Quantum Information and Measurement (QIM) 2017*, page QF6A.2. Optical Society of America, 2017.

[71] J. I. Colless, V. V. Ramasesh, D. Dahlen, M. S. Blok, M. E. Kimchi-Schwartz, J. R. McClean, J. Carter, W. A. de Jong, and I. Siddiqi. Computation of molecular spectra on a quantum processor with an error-resilient algorithm. *Phys. Rev. X*, **8**, 011021 (2018).

[72] B. Cooper and P. J. Knowles. Benchmark studies of variational, unitary and extended coupled cluster methods. *J. Chem. Phys.*, **133**, 234102 (2010).

[73] L. N. Cooper. Bound electron pairs in a degenerate fermi gas. *Phys. Rev.*, **104**, 1189 (1956).

[74] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath. Generative adversarial networks: An overview. *IEEE Signal Process. Mag.*, **35**, 53 (2018).

[75] G. E. Crooks. Performance of the quantum approximate optimization algorithm on the maximum cut problem. arXiv preprint arXiv:1811.08419 (2018).

[76] S. Curtarolo, G. L. W. Hart, M. B. Nardelli, N. Mingo, S. Sanvito, and O. Levy. The high-throughput highway to computational materials design. *Nat. Mater.*, **12**, 191 (2013).

[77] P.-L. Dallaire-Demers and N. Killoran. Quantum generative adversarial networks. *Phys. Rev. A*, **98**, 012324 (2018).

[78] P.-L. Dallaire-Demers, J. Romero, L. Veis, S. Sim, and A. Aspuru-Guzik. Low-depth circuit ansatz for preparing correlated fermionic states on a quantum computer. arXiv preprint arXiv:1801.01053 (2018).

[79] P.-L. Dallaire-Demers and F. K. Wilhelm. Method to efficiently simulate the thermodynamic properties of the fermi-Hubbard model on a quantum computer. *Phys. Rev. A*, **93**, 032303 (2016).

[80] P.-L. Dallaire-Demers and F. K. Wilhelm. Quantum gates and architecture for the quantum simulation of the fermi-Hubbard model. *Phys. Rev. A*, **94**, 062304 (2016).

[81] M. Dalmonte and S. Montangero. Lattice gauge theory simulations in the quantum information era. *Contemp. Phys.*, **57**, 388 (2016).

[82] C. Dankert, R. Cleve, J. Emerson, and E. Livine. Exact and approximate unitary 2-designs and their application to fidelity estimation. *Phys. Rev. A*, **80**, 012304 (2009).

[83] A. Daskin, A. Grama, G. Kollias, and S. Kais. Universal programmable quantum circuit schemes to emulate an operator. *J. Chem. Phys.*, **137**, 234112 (2012).

[84] N. Datta, J. M. Renes, R. Renner, and M. M. Wilde. One-shot lossy quantum data compression. *IEEE Trans. Inf. Theory*, **59**, 8057 (2013).

[85] E. R. Davidson. Properties and uses of natural orbitals. *Rev. Mod. Phys.*, **44**, 451 (1972).

[86] S. Debnath, N. M. Linke, C. Figgatt, K. A. Landsman, K. Wright, and C. R. Monroe. Demonstration of a small programmable quantum computer with atomic qubits. *Nature*, **536**, 63 (2016).

[87] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill. Topological quantum memory. *J. Math. Phys.*, **43**, 4452 (2002).

[88] D. Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proc. R. Soc. Lond. A*, **400**, 97 (1985).

[89] D. E. Deutsch. Quantum computational networks. *Proc. R. Soc. Lond. A*, **425**, 73 (1989).

[90] M. H. Devoret and R. J. Schoelkopf. Superconducting circuits for quantum information: An outlook. *Science*, **339**, 1169 (2013).

[91] L. DiCarlo, J. Chow, J. Gambetta, L. S. Bishop, B. Johnson, D. Schuster, J. Majer, A. Blais, L. Frunzio, S. Girvin, et al. Demonstration of two-qubit algorithms with a superconducting quantum processor. *Nature*, **460**, 240 (2009).

[92] D. P. DiVincenzo. The physical implementation of quantum computation. *Fortschritte der Physik: Progress of Physics*, **48**, 771 (2000).

[93] D. P. DiVincenzo. The physical implementation of quantum computation. *Fortschr. Phys.*, **48**, 771 (2000).

[94] J. Du, N. Xu, X. Peng, P. Wang, S. Wu, and D. Lu. NMR implementation of a molecular hydrogen quantum simulation with adiabatic state preparation. *Phys. Rev. Lett.*, **104**, 030502 (2010).

[95] Y. Du, M.-H. Hsieh, T. Liu, and D. Tao. The expressive power of parameterized quantum circuits. arXiv preprint arXiv:1810.11922 (2018).

[96] F. Dupuis, M. Berta, J. Wullschleger, and R. Renner. One-shot decoupling. *Commun. Math. Phys.*, **328**, 251 (2014).

[97] J. Emerson, R. Alicki, and K. Życzkowski. Scalable noise estimation with random unitary operators. *J. Opt. B: Quantum Semiclassical Opt.*, **7**, S347 (2005).

[98] S. Endo, S. C. Benjamin, and Y. Li. Practical quantum error mitigation for near-future applications. *Phys. Rev. X*, **8**, 031027 (2018).

[99] S. Endo, T. Jones, S. McArdle, X. Yuan, and S. Benjamin. Variational quantum algorithms for discovering hamiltonian spectra. arXiv preprint arXiv:1806.05707 (2018).

[100] S. Endo, Y. Li, S. Benjamin, and X. Yuan. Variational quantum simulation of general processes. arXiv preprint arXiv:1812.08778 (2018).

[101] S. Er, C. Suh, M. P. Marshak, and A. Aspuru-Guzik. Computational design of molecules for an all-quinone redox flow battery. *Chem. Sci.*, **6**, 885 (2015).

[102] F. H. L. Essler, H. Frahm, F. Gohmann, A. Klumper, and V. E. Korepin. *The one-dimensional Hubbard model*. Cambridge University Press, 2005.

[103] F. A. Evangelista. Alternative single-reference coupled cluster approaches for multireference problems: The simpler, the better. *J. Chem. Phys.*, **134**, 224102 (2011).

[104] E. Farhi, J. Goldstone, and S. Gutmann. A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028 (2014).

[105] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. Quantum computation by adiabatic evolution. arXiv preprint arXiv:quant-ph/0001106 (2000).

[106] E. Farhi and A. W. Harrow. Quantum supremacy through the quantum approximate optimization algorithm. arXiv preprint arXiv:1602.07674 (2016).

[107] E. Farhi and H. Neven. Classification with quantum neural networks on near term processors. arXiv preprint arXiv:1802.06002 (2018).

[108] R. P. Feynman. Simulating physics with computers. *Int. J. Theor. Phys.*, **21**, 467 (1982).

[109] A. S. Fletcher, P. W. Shor, and M. Z. Win. Channel-adapted quantum error correction for the amplitude damping channel. *IEEE Trans. Inf. Theory*, **54**, 5705 (2008).

[110] A. S. Fletcher, P. W. Shor, and M. Z. Win. Structured near-optimal channel-adapted quantum error correction. *Phys. Rev. A*, **77**, 012320 (2008).

[111] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A.*, **86**, 032324 (2012).

[112] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A*, **86**, 032324 (2012).

[113] J. M. Gambetta, J. M. Chow, and M. Steffen. Building logical qubits in a superconducting quantum computing system. *npj Quantum Inf.*, **3**, 2 (2017).

[114] S. Gammelmark and K. Mølmer. Quantum learning by measurement and feedback. *New. J. Phys*, **11**, 033017 (2009).

[115] I. M. Gelfand, R. A. Silverman, et al. *Calculus of variations*. Courier Corporation, 2000.

[116] M. R. Geller and Z. Zhou. Efficient error models for fault-tolerant architectures and the Pauli twirling approximation. *Phys. Rev. A*, **88**, 012314 (2013).

[117] I. Georgescu, S. Ashhab, and F. Nori. Quantum simulation. *Rev. Mod. Phys.*, **86**, 153 (2014).

[118] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, **4**, 268 (2018).

[119] I. Goodfellow. Nips 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160 (2016).

[120] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[121] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems 27*, pages 2672–2680, 2014.

[122] D. Gottesman. Stabilizer codes and quantum error correction. arXiv preprint quant-ph/9705052 (1997).

[123] E. Grant, M. Benedetti, S. Cao, A. Hallam, J. Lockhart, V. Stojevic, A. G. Green, and S. Severini. Hierarchical quantum classifiers. *npj Quantum Inf.*, **4**, 65 (2018).

[124] M. Grant, S. Boyd, and Y. Ye. Cvx: Matlab software for disciplined convex programming, 2008.

[125] M. C. Grant and S. P. Boyd. Graph implementations for nonsmooth convex programs. In V. D. Blondel, S. P. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, pages 95–110, London, 2008. Springer London.

[126] D. J. Griffiths and D. F. Schroeter. *Introduction to quantum mechanics*. Cambridge University Press, 2018.

[127] L. Grover and T. Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. arXiv preprint quant-ph/0208112 (2002).

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

[129] E. Grumbling and M. Horowitz, editors. *Quantum Computing: Progress and Prospects*. The National Academies Press, Washington, DC, 2018.

[130] G. Guerreschi and A. Matsuura. Qaoa for max-cut requires hundreds of qubits for quantum speed-up. arXiv preprint arXiv:1812.07589 (2018).

[131] G. G. Guerreschi and M. Smelyanskiy. Practical optimization for hybrid quantum-classical algorithms. arXiv preprint arXiv:1701.01450 (2017).

[132] M. Gutiérrez and K. R. Brown. Comparison of a quantum error-correction threshold for exact and approximate errors. *Phys. Rev. A*, **91**, 022335 (2015).

[133] J. Hachmann, R. Olivares-Amaya, S. Atahan-Evrenk, C. Amador-Bedolla, R. S. Sánchez-Carrera, A. Gold-Parker, L. Vogt, A. M. Brockway, and A. Aspuru-Guzik. The Harvard clean energy project: large-scale computational screening and design of organic photovoltaics on the world community grid. *J. Phys. Chem. Lett.*, **2**, 2241 (2011).

[134] H. Häffner, C. F. Roos, and R. Blatt. Quantum computing with trapped ions. *Phys. Rep.*, **469**, 155 (2008).

[135] D. Hanneke, J. P. Home, J. D. Jost, J. M. Amini, D. Leibfried, and D. J. Wineland. Realization of a programmable two-qubit quantum processor. *Nat. Phys.*, **6**, 13 (2010).

[136] S. Haroche. Nobel lecture: Controlling photons in a box and exploring the quantum to classical boundary. *Rev. Mod. Phys.*, **85**, 1083 (2013).

[137] M. B. Hastings, D. Wecker, B. Bauer, and M. Troyer. Improving quantum algorithms for quantum chemistry. *Quantum Inf. Comput.*, **15**, 1 (2015).

[138] V. Havlicek, A. D. Córcoles, K. Temme, A. W. Harrow, J. M. Chow, and J. M. Gambetta. Supervised learning with quantum enhanced feature spaces. arXiv preprint arXiv:1804.11326 (2018).

[139] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 770–778, 2016.

[140] M. Head-Gordon and E. A. Cortés. Chemistry on the computer. *Phys. Today*, **61**, 58 (2008).

[141] T. Helgaker, J. Olsen, and P. Jorgensen. *Molecular electronic structure theory*. Wiley, 2013.

[142] C. Hempel, C. Maier, J. Romero, J. McClean, T. Monz, H. Shen, P. Jurcevic, B. P. Lanyon, P. Love, R. Babbush, A. Aspuru-Guzik, R. Blatt, and C. F. Roos. Quantum chemistry calculations on a trapped-ion quantum simulator. *Phys. Rev. X*, **8**, 031022 (2018).

[143] U. L. Heras, A. Mezzacapo, L. Lamata, S. Filipp, A. Wallraff, and E. Solano. Digital quantum simulation of spin systems in superconducting circuits. *Phys. Rev. Lett.*, **112**, 200501 (2014).

[144] F. Herrera, Y. Cao, S. Kais, and K. B. Whaley. Infrared-dressed entanglement of cold open-shell polar molecules for universal matchgate quantum computing. *New J. Phys.*, **16**, 075001 (2014).

[145] K. Heya, Y. Suzuki, Y. Nakamura, and K. Fujii. Variational quantum gate optimization. arXiv preprint arXiv:1810.12745 (2018).

[146] O. Higgott, D. Wang, and S. Brierley. Variational quantum computation of excited states. arXiv preprint arXiv:1805.08138 (2018).

[147] C. D. Hill, E. Peretz, S. J. Hile, M. G. House, M. Fuechsle, S. Rogge, M. Y. Simmons, and L. C. L. Hollenberg. A surface code quantum computer in silicon. *Sci. Adv.*, **1**, 1 (2015).

[148] A. Ho, J. McClean, and S. P. Ong. The promise and challenges of quantum computing for energy storage. *Joule*, **2**, 810 (2018).

[149] D. K. Hoffman, R. C. Raffenetti, and K. Ruedenberg. Generalization of euler angles to nâĂŘdimensional orthogonal matrices. *J. Math. Phys.*, **13**, 528 (1972).

[150] M. R. Hoffmann and J. Simons. A unitary multiconfigurational coupled-cluster method: Theory and applications. *J. Chem. Phys.*, **88**, 993 (1988).

[151] L. Hu, S.-H. Wu, W. Cai, Y. Ma, X. Mu, Y. Xu, H. Wang, Y. Song, D.-L. Deng, C.-L. Zou, et al. Quantum generative adversarial learning in a superconducting quantum circuit. arXiv preprint arXiv:1808.02893 (2018).

[152] J. Hubbard. Electron correlations in narrow energy bands. *Proc. R. Soc. A*, **276**, 238 (1963).

[153] W. J. Huggins, P. Patil, B. Mitchell, K. B. Whaley, and M. Stoudenmire. Towards quantum machine learning with tensor networks. arXiv preprint arXiv:1803.11537 (2018).

[154] B. Huskinson, M. P. Marshak, C. Suh, S. Er, M. R. Gerhardt, C. J. Galvin, X. Chen, A. Aspuru-Guzik, R. G. Gordon, and M. J. Aziz. A metal-free organic-inorganic aqueous flow battery. *Nature*, **505**, 195 (2014).

[155] F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *International Conference on Learning and Intelligent Optimization*, pages 507–523. Springer, 2011.

[156] A. Imamog, D. D. Awschalom, G. Burkard, D. P. DiVincenzo, D. Loss, M. Sherwin, A. Small, et al. Quantum information processing using quantum dot spins and cavity QED. *Phys. Rev. Lett.*, **83**, 4204 (1999).

[157] M. Jaderberg, V. Dalibard, S. Osindero, W. Czarnecki, J. Donahue, A. Razavi, O. Vinyals, T. Green, I. Dunning, K. Simonyan, C. Fernando, and K. Kavukcuoglu. Population based training of neural networks. *CoRR*, abs/1711.09846 (2017).

[158] K. Jankowski and J. Paldus. Applicability of coupled-pair theories to quasidegenerate electronic states: A model study. *Int. J. Quantum Chem.*, **18**, 1243 (1980).

[159] H. J. A. Jensen, P. Jørgensen, H. Ågren, and J. Olsen. Second-order møller-plesset perturbation theory as a configuration and orbital generator in multiconfiguration self-consistent field calculations. *J. Chem. Phys*, **88**, 3834 (1988).

[160] Z. Jiang, K. J. Sung, K. Kechedzhi, V. N. Smelyanskiy, and S. Boixo. Quantum algorithms to simulate many-body physics of correlated fermions. *Phys. Rev. Appl*, **9**, 044036 (2018).

[161] J. R. Johansson, P. D. Nation, and F. Nori. Qutip: An open-source python framework for the dynamics of open quantum systems. *Comput. Phys. Commun.*, **183**, 1760 (2012).

[162] J. R. Johansson, P. D. Nation, and F. Nori. Qutip 2: A python framework for the dynamics of open quantum systems. *Comput. Phys. Commun.*, **184**, 1234 (2013).

[163] P. D. Johnson, J. Romero, J. Olson, Y. Cao, and A. Aspuru-Guzik. Qvector: an algorithm for device-tailored quantum error correction. arXiv.org preprint arXiv:1711.02249 (2017).

[164] P. Jordan and E. Wigner. Über das Paulische Äquivalenzverbot. *Zeitschrift für Physik*, **47**, 631 (1928).

[165] P. Jordan and E. P. Wigner. About the Pauli exclusion principle. *Z. Angew. Phys.*, **47**, 631 (1928).

[166] R. Jozsa and N. Linden. On the role of entanglement in quantum-computational speed-up. *Proc. R. Soc. Lond. A*, **459**, 2011 (2003).

[167] R. Jozsa and A. Miyake. Matchgates and classical simulation of quantum circuits. *Proc. R. Soc. A*, **464**, 3089 (2008).

[168] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, **549**, 242 (2017).

[169] A. Kandala, K. Temme, A. D. Corcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta. Extending the computational reach of a noisy superconducting quantum processor. arXiv preprint arXiv:1805.04492 (2018).

[170] I. Kassal, S. P. Jordan, P. J. Love, M. Mohseni, and A. Aspuru-Guzik. Polynomial-time quantum algorithm for the simulation of chemical dynamics. *PNAS* (2008).

[171] I. Kassal, J. D. Whitfield, A. Perdomo-Ortiz, M.-H. Yung, and A. Aspuru-Guzik. Simulating chemistry using quantum computers. *Annu. Rev. Phys. Chem.*, **62**, 185 (2011). PMID: 21166541.

[172] A. Katabarwa. A dynamical interpretation of the Pauli twirling approximation and quantum error correction. arXiv preprint arXiv:1701.03708 (2017).

[173] A. Katabarwa and M. R. Geller. Logical error rate in the Pauli twirling approximation. *Sci. Rep.*, **5**, 14670 (2015).

[174] S. Keller, K. Boguslawski, T. Janowski, M. Reiher, and P. Pulay. Selection of active spaces for multiconfigurational wavefunctions. *J. Chem. Phys*, **142**, 244104 (2015).

[175] J. Kelly, R. Barends, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, I.-C. Hoi, E. Jeffrey, A. Megrant, J. Mutus, C. Neill, P. J. J. O'Malley, C. Quintana, P. Roushan, D. Sank, A. Vainsencher, J. Wenner, T. C. White, A. N. Cleland, and J. M. Martinis. Optimal quantum control using randomized benchmarking. *Phys. Rev. Lett.*, **112**, 240504 (2014).

[176] J. Kelly, R. Barends, A. G. Fowler, A. Megrant, E. Jeffrey, T. C. White, D. Sank, J. Y. Mutus, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, I. C. Hoi, C. Neill, P. J. J. O'Malley, C. Quintana, P. Roushan, A. Vainsencher, J. Wenner, A. N. Cleland, and J. M. Martinis. State preservation by repetitive error detection in a superconducting quantum circuit. *Nature*, **519**, 66 (2015).

[177] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-HerbrÃijggen, and S. J. Glaser. Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms. *J. Magn. Reson.*, **172**, 296 (2005).

[178] A. Y. Kitaev, A. Shen, and M. N. Vyalyi. *Classical and Quantum Computation*. American Mathematical Society, Boston, MA, USA, 2002.

[179] I. D. Kivlichan, J. McClean, N. Wiebe, C. Gidney, A. Aspuru-Guzik, G. K.-L. Chan, and R. Babbush. Quantum simulation of electronic structure with linear depth and connectivity. *Phys. Rev. Lett.*, **120**, 110501 (2018).

[180] E. Knill. Protected realizations of quantum information. *Phys. Rev. A*, **74**, 042301 (2006).

[181] E. Knill and R. Laflamme. Theory of quantum error-correcting codes. *Phys. Rev. A*, **55**, 900 (1997).

[182] J. Koch, T. M. Yu, J. Gambetta, A. A. Houck, D. I. Schuster, J. Majer, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf. Charge-insensitive qubit design derived from the cooper pair box. *Phys. Rev. A*, **76**, 042319 (2007).

[183] T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM. Rev*, **45**, 385 (2003).

[184] R. L. Kosut and D. A. Lidar. Quantum error correction via convex optimization. *Quantum Inf. Process.*, **8**, 443 (2009).

[185] R. L. Kosut, A. Shabani, and D. A. Lidar. Robust quantum error correction via convex optimization. *Phys. Rev. Lett.*, **100**, 020502 (2008).

[186] B. Kraus and J. I. Cirac. Optimal creation of entanglement using a two-qubit gate. *Phys. Rev. A*, **63**, 062309 (2001).

[187] C. V. Kraus and J. I. Cirac. Generalized Hartree-Fock theory for interacting fermions in lattices: numerical methods. *New J. Phys.*, **12**, 113004 (2010).

[188] C. V. Kraus, M. M. Wolf, J. I. Cirac, and G. Giedke. Pairing in fermionic systems: A quantum-information perspective. *Phys. Rev. A*, **79**, 012306 (2009).

[189] W. Kutzelnigg. *Pair Correlation Theories*, pages 129–188. Springer US, Boston, MA, 1977.

[190] W. Kutzelnigg. Error analysis and improvements of coupled-cluster theory. *Theor. Chim. Acta*, **80**, 349 (1991).

[191] M. Laforest, D. Simon, J.-C. Boileau, J. Baugh, M. J. Ditty, and R. Laflamme. Using error correction to determine the noise model. *Phys. Rev. A*, **75**, 012331 (2007).

[192] B. P. Lanyon, C. Hempel, D. Nigg, M. Müller, R. Gerritsma, F. Zähringer, P. Schindler, J. T. Barreiro, M. Rambach, G. Kirchmair, M. Hennrich, P. Zoller, R. Blatt, and C. F. Roos. Universal digital quantum simulation with trapped ions. *Science*, **334**, 57 (2011).

[193] R. LaRose, A. Tikku, É. O'Neel-Judy, L. Cincio, and P. J. Coles. Variational quantum state diagonalization. arXiv preprint arXiv:1810.10506 (2018).

[194] U. Las Heras, L. García-Álvarez, A. Mezzacapo, E. Solano, and L. Lamata. Fermionic models with superconducting circuits. *EPJ Quantum Technology*, **2**, 8 (2015).

[195] D. Layden and P. Cappellaro. Error-corrected quantum sensing with parallel signal and noise. arXiv preprint arXiv:1708.06729 (2017).

[196] A. J. Leggett. A "midinfrared" scenario for cuprate superconductivity. *PNAS*, **96**, 8365 (1999).

[197] A. J. Leggett. *Quantum Liquids*. Oxford University Press, 2006.

[198] D. Leibfried, B. DeMarco, V. Meyer, D. Lucas, M. Barrett, J. Britton, W. M. Itano, B. Jelenković, C. Langer, T. Rosenband, et al. Experimental demonstration of a robust, high-fidelity geometric two ion-qubit phase gate. *Nature*, **422**, 412 (2003).

[199] D. Leibfried, B. DeMarco, V. Meyer, D. Lucas, M. Barrett, J. Britton, W. M. Itano, B. Jelenkovic, C. Langer, T. Rosenband, and D. J. Wineland. Experimental demonstration of a robust, high-fidelity geometric two ion-qubit phase gate. *Nature*, **422**, 412 (2003).

[200] D. W. Leung, M. A. Nielsen, I. L. Chuang, and Y. Yamamoto. Approximate quantum error correction can lead to better codes. *Phys. Rev. A*, **56**, 2567 (1997).

[201] J. Li, X. Yang, X. Peng, and C.-P. Sun. Hybrid quantum-classical approach to quantum optimal control. *Phys. Rev. Lett.*, **118**, 150503 (2017).

[202] Y. Li and S. C. Benjamin. Efficient variational quantum simulator incorporating active error minimization. *Phys. Rev. X*, **7**, 021050 (2017).

[203] D. A. Lidar and T. A. Brun. *Quantum error correction*. Cambridge University Press, 2013.

[204] E. H. Lieb and F. Y. Wu. The one-dimensional Hubbard model: A reminiscence. *Physica A*, **321**, 1 (2003).

[205] N. M. Linke, M. Gutierrez, K. A. Landsman, C. Figgatt, S. Debnath, K. R. Brown, and C. Monroe. Fault-tolerant quantum error detection. *Sci. Adv.*, **3**, 1 (2017).

[206] N. M. Linke, D. Maslov, M. Roetteler, S. Debnath, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe. Experimental comparison of two quantum computing architectures. *PNAS*, **114**, 3305 (2017).

[207] C.-Y. Liou, W.-C. Cheng, J.-W. Liou, and D.-R. Liou. Autoencoder for words. *Neurocomputing*, **139**, 84 (2014).

[208] C.-Y. Liou, J.-C. Huang, and W.-C. Yang. Modeling word perception using the elman network. *Neurocomputing*, **71**, 3150 (2008).

[209] J.-G. Liu and L. Wang. Differentiable learning of quantum circuit born machines. *Phys. Rev. A*, **98**, 062324 (2018).

[210] S. Lloyd. Universal quantum simulators. *Science*, **273**, 1073 (1996).

[211] S. Lloyd. Quantum approximate optimization is computationally universal. arXiv preprint arXiv:1812.11075 (2018).

[212] S. Lloyd and C. Weedbrook. Quantum generative adversarial learning. *Phys. Rev. Lett.*, **121**, 040502 (2018).

[213] B. A. Loiseau and Y. Nogami. Three-nucleon force. *Nucl. Phys. B*, **2**, 470 (1967).

[214] D. I. Lyakh, M. Musial, V. F. Lotrich, and R. J. Bartlett. Multireference nature of chemistry: The coupled-cluster view. *Chem. Rev.*, **112**, 182 (2011).

[215] S. Machnes, U. Sander, S. J. Glaser, P. de Fouquières, A. Gruslys, S. Schirmer, and T. Schulte-Herbrüggen. Comparing, optimizing, and benchmarking quantum-control algorithms in a unifying programming framework. *Phys. Rev. A*, **84**, 022305 (2011).

[216] E. Magesan, J. M. Gambetta, B. R. Johnson, C. A. Ryan, J. M. Chow, S. T. Merkel, M. P. da Silva, G. A. Keefe, M. B. Rothwell, T. A. Ohki, M. B. Ketchen, and M. Steffen. Efficient measurement of quantum gate error by interleaved randomized benchmarking. *Phys. Rev. Lett.*, **109**, 080505 (2012).

[217] Y. Manin. Computable and uncomputable (in russian). *Sovetskoye Radio, Moscow* (1980).

[218] N. Margolus. Parallel quantum computation. *Complexity, Entropy and the Physics of Information, Santa Fe Institute Studies in the Sciences of Complexity*, **8**, 273 (1990).

[219] K. H. Marti and M. Reiher. New electron correlation theories for transition metal chemistry. *Phys. Chem. Chem. Phys.*, **13**, 6750 (2011).

[220] N. Mataga and K. Nishimoto. Electronic structure and spectra of nitrogen heterocycles. *Zeitschrift für Physikalische Chemie*, **13**, 140 (1957).

[221] S. McArdle, X. Yuan, and S. Benjamin. Error-mitigated digital quantum simulation. *ArXiv e-prints* (2018).

[222] J. R. McClean, R. Babbush, P. J. Love, and A. Aspuru-Guzik. Exploiting locality in quantum computation for quantum chemistry. *J. Phys. Chem. Lett.*, **5**, 4368 (2014).

[223] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven. Barren plateaus in quantum neural network training landscapes. *Nat. Commun.*, **9**, 4812 (2018).

[224] J. R. McClean, M. E. Kimchi-Schwartz, J. Carter, and W. A. de Jong. Hybrid quantum-classical hierarchy for mitigation of decoherence and determination of excited states. *Phys. Rev. A*, **95**, 042308 (2017).

[225] J. R. McClean, I. D. Kivlichan, D. S. Steiger, Y. Cao, E. S. Fried, C. Gidney, T. Häner, V. Havlíček, Z. Jiang, M. Neeley, et al. OpenFermion: The electronic structure package for quantum computers. arXiv preprint arXiv:1710.07629 (2017).

[226] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New J. Phys.*, **18**, 023023 (2016).

[227] J. T. Merrill and K. R. Brown. *Progress in compensating pulse sequences for quantum computation*, pages 241–294. John Wiley & Sons, Ltd, 2014.

[228] L. Mescheder, A. Geiger, and S. Nowozin. Which training methods for GANs do actually converge? In *Proceedings of Machine Learning Research*, pages 3478–3487, 2018.

[229] A. Mezzacapo, L. Lamata, S. Filipp, and E. Solano. Many-body interactions with tunable-coupling transmon qubits. *Phys. Rev. Lett.*, **113**, 050501 (2014).

[230] R. Miikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, H. Shahrzad, A. Navruzyan, N. Duffy, et al. Evolving deep neural networks. In *Artificial Intelligence in the Age of Neural Networks and Brain Computing*, pages 293–312. Elsevier, 2019.

[231] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii. Quantum circuit learning. *Phys. Rev. A*, **98**, 032309 (2018).

[232] J. Mockus. *Bayesian approach to global optimization: theory and applications*, volume 37. Springer Science & Business Media, 2012.

[233] M. Mohseni, P. Read, H. Neven, S. Boixo, V. Denchev, R. Babbush, A. Fowler, V. Smelyanskiy, and J. Martinis. Commercialize quantum technologies in five years. *Nature News*, **543**, 171 (2017).

[234] N. Moll, P. Barkoutsos, L. S. Bishop, J. M. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn, A. Kandala, A. Mezzacapo, P. MÃijller, W. Riess, G. Salis, J. Smolin, I. Tavernelli, and K. Temme. Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Sci. Technol.*, **3**, 030503 (2018).

[235] C. R. Monroe and J. Kim. Scaling the ion trap quantum processor. *Science*, **339**, 1164 (2013).

[236] A. Montanaro. Quantum algorithms: an overview. *npj Quantum Inf.*, **2**, 15023 (2016).

[237] T. Monz, D. Nigg, E. A. Martinez, M. F. Brandl, P. Schindler, R. Rines, S. X. Wang, I. L. Chuang, and R. Blatt. Realization of a scalable shor algorithm. *Science*, **351**, 1068 (2016).

[238] M. Müller, K. Hammerer, Y. L. Zhou, C. F. Roos, and P. Zoller. Simulating open quantum systems: from many-body interactions to stabilizer pumping. *New J. Phys.*, **13**, 085007 (2011).

[239] Y. Nakata, C. Hirche, C. Morgan, and A. Winter. Unitary 2-designs from random x-and z-diagonal unitaries. *J. Math. Phys.*, **58**, 052203 (2017).

[240] R. D. Neidinger. Introduction to automatic differentiation and matlab object-oriented programming. *SIAM Rev.*, **52**, 545 (2010).

[241] C. Neill, P. Roushan, K. Kechedzhi, S. Boixo, S. V. Isakov, V. Smelyanskiy, A. Megrant, B. Chiaro, A. Dunsworth, K. Arya, R. Barends, B. Burkett, Y. Chen, Z. Chen, A. Fowler, B. Foxen, M. Giustina, R. Graff, E. Jeffrey, T. Huang, J. Kelly, P. Klimov, E. Lucero, J. Mutus, M. Neeley, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, H. Neven, and J. M. Martinis. A blueprint for demonstrating quantum supremacy with superconducting qubits. *Science*, **360**, 195 (2018).

[242] J. A. Nelder and R. Mead. A simplex method for function minimization. *Comput. J.*, **7**, 308 (1965).

[243] M. A. Nielsen and I. L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2010.

[244] P. Niemann, R. Datta, and R. Wille. Logic synthesis for quantum state generation. In *2016 IEEE 46th International Symposium on Multiple-Valued Logic (ISMVL)*, pages 247–252, May 2016.

[245] D. Nigg, M. Müller, E. A. Martinez, P. Schindler, M. Hennrich, T. Monz, M. A. Martin-Delgado, and R. Blatt. Quantum computations on a topologically encoded qubit. *Science* (2014).

[246] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, second edition, 2006.

[247] J. L. O'Brien, G. J. Pryde, A. G. White, T. C. Ralph, and D. Branning. Demonstration of an all-optical quantum controlled-not gate. *Nature*, **426**, 264 (2003).

[248] P. J. J. O'Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, E. Jeffrey, E. Lucero, A. Megrant, J. Y. Mutus, M. Neeley, C. Neill, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, P. V. Coveney, P. J. Love, H. Neven, A. Aspuru-Guzik, and J. M. Martinis. Scalable quantum simulation of molecular energies. *Phys. Rev. X*, **6**, 031007 (2016).

[249] G. Ortiz, J. Gubernatis, E. Knill, and R. Laflamme. Quantum algorithms for fermionic simulations. *Phys. Rev. A*, **64**, 022319 (2001).

[250] R. Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Ann. Phys.*, **349**, 117 (2014).

[251] R. Pariser and R. G. Parr. A semi-empirical theory of the electronic spectra and electronic structure of complex unsaturated molecules. i. *J. Chem. Phys.*, **21**, 466 (1953).

[252] R. Pariser and R. G. Parr. A semi-empirical theory of the electronic spectra and electronic structure of complex unsaturated molecules. II. *J. Chem. Phys.*, **21**, 767 (1953).

[253] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *NIPS-W*, 2017.

[254] A. Perdomo-Ortiz, M. Benedetti, J. Realpe-Gómez, and R. Biswas. Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers. *Quantum Sci. Technol.*, **3**, 030502 (2018).

[255] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.*, **5**, 4213 (2014).

[256] M. J. Peterer, S. J. Bader, X. Jin, F. Yan, A. Kamal, T. J. Gudmundsen, P. J. Leek, T. P. Orlando, W. D. Oliver, and S. Gustavsson. Coherence and decay of higher energy levels of a superconducting transmon qubit. *Phys. Rev. Lett.*, **114**, 010501 (2015).

[257] K. A. Peterson, D. Feller, and D. A. Dixon. Chemical accuracy in ab initio thermochemistry and spectroscopy: current strategies and future challenges. *Theor. Chem. Acc.*, **131**, 1 (2012).

[258] J. Plantenberg, P. De Groot, C. Harmans, and J. Mooij. Demonstration of controlled-not quantum gates on a pair of superconducting quantum bits. *Nature*, **447**, 836 (2007).

[259] M. Plesch and i. c. v. Brukner. Quantum-state preparation with universal gate decompositions. *Phys. Rev. A*, **83**, 032302 (2011).

[260] C. Pomerance. A tale of two sieves. *Notices Amer. Math. Soc*, **43**, 1473 (1996).

[261] J. A. Pople. Electron interaction in unsaturated hydrocarbons. *Trans. Faraday Soc.*, **49**, 1375 (1953).

[262] M. Potthoff. Non-perturbative construction of the luttinger-ward functional. *Condens. Matter Phys.*, **9**, 557 (2006).

[263] M. Potthoff, M. Aichhorn, and C. Dahnken. Variational cluster approach to correlated electron systems in low dimensions. *Phys. Rev. Lett.*, **91**, 206402 (2003).

[264] D. Poulin. Stabilizer formalism for operator quantum error correction. *Phys. Rev. Lett.*, **95**, 230504 (2005).

[265] D. Poulin, M. B. Hastings, D. Wecker, N. Wiebe, A. C. Doherty, and M. Troyer. The trotter step size required for accurate quantum simulation of quantum chemistry. *Quantum Inf. Comput.*, **15**, 361 (2015).

[266] M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, **7**, 155 (1964).

[267] M. J. D. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In S. Gomez and J.-P. Hennart, editors, *Advances in Optimization and Numerical Analysis*, pages 51–67. Springer Netherlands, Dordrecht, 1994.

[268] J. Preskill. Quantum computing and the entanglement frontier. arXiv preprint arXiv:1203.5813 (2012).

[269] J. Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, **2**, 79 (2018).

[270] Quantum information software kit (qiskit). `https://qiskit.org/`, 2018.

[271] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015).

[272] M. Reiher, N. Wiebe, K. M. Svore, D. Wecker, and M. Troyer. Elucidating reaction mechanisms on quantum computers. *PNAS* (2017).

[273] M. Reimpell and R. F. Werner. Iterative optimization of quantum error correcting codes. *Phys. Rev. Lett.*, **94**, 080501 (2005).

[274] G. Rickayzen. *Green's Functions and Condensed Matter*. Academic Press, 1991.

[275] P. Ring and P. Schuck. *The nuclear many-body problem*. Springer, 1980.

[276] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, **21**, 120 (1978).

[277] Z. Rolik and M. Kállay. A quasiparticle-based multi-reference coupled-cluster method. *J. Chem. Phys.*, **141**, 134112 (2014).

[278] J. Romero and A. Aspuru-Guzik. Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions. arXiv preprint arXiv:1901.00848 (2019).

[279] J. Romero, R. Babbush, J. R. McClean, C. Hempel, P. J. Love, and A. Aspuru-Guzik. Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz. *Quantum Sci. Technol.*, **4**, 014008 (2018).

[280] J. Romero, J. P. Olson, and A. Aspuru-Guzik. Quantum autoencoders for efficient compression of quantum data. *Quantum Sci. Technol.*, **2**, 045001 (2017).

[281] B. O. Roos, P. R. Taylor, and P. E. M. Siegbahn. A complete active space SCF method (CASSCF) using a density matrix formulated super-CI approach. *Chem. Phys.*, **48**, 157 (1980).

[282] K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann. Stabilizing training of generative adversarial networks through regularization. In *Advances in Neural Information Processing Systems 30*, pages 2018–2028, 2017.

[283] N. Rubin, R. Babbush, and J. McClean. Application of fermionic marginal constraints to hybrid quantum algorithms. *New J. Phys.*, **20**, 053020 (2018).

[284] C. A. Sackett, D. Kielpinski, B. E. King, C. Langer, V. Meyer, C. J. Myatt, M. Rowe, Q. A. Turchette, W. M. Itano, D. J. Wineland, et al. Experimental entanglement of four particles. *Nature*, **404**, 256 (2000).

[285] B. C. Sanders. How to build a quantum computer. In *How to Build a Quantum Computer*, 2399-2891, pages 1–1 to 1–15. IOP Publishing, 2017.

[286] Y. R. Sanders, J. J. Wallman, and B. C. Sanders. Bounding quantum gate error rate based on reported average fidelity. *New J. Phys.*, **18**, 012002 (2016).

[287] R. Santagati, J. Wang, A. A. Gentile, S. Paesani, N. Wiebe, J. R. McClean, S. Morley-Short, P. J. Shadbolt, D. Bonneau, J. W. Silverstone, D. P. Tew, X. Zhou, J. L. O'Brien, and M. G. Thompson. Witnessing eigenstates for quantum simulation of hamiltonian spectra. *Sci. Adv.*, 4 (2018).

[288] N. P. D. Sawaya, M. Smelyanskiy, J. R. McClean, and A. Aspuru-Guzik. Error sensitivity to environmental noise in quantum circuits for chemical state preparation. *J. Chem. Theory Comput.*, **12**, 3097 (2016). PMID: 27254482.

[289] P. Schindler, D. Nigg, T. Monz, J. T. Barreiro, E. Martinez, S. X. Wang, S. Quint, M. F. Brandl, V. Nebendahl, C. F. Roos, M. Chwalla, M. Hennrich, and R. Blatt. A quantum information processor with trapped ions. *New J. Phys.*, **15**, 123012 (2013).

[290] T. G. Schmalz. From the Hubbard to the PPP model. *Croat. Chem. Acta*, **86**, 419 (2013).

[291] T. G. Schmalz, L. Serrano-Andrés, V. Sauri, M. Merchán, and J. M. Oliva. A distance-dependent parameterization of the extended Hubbard model for conjugated and aromatic hydrocarbons derived from stretched ethene. *J. Chem. Phys.*, **135**, 194103 (2011).

[292] M. W. Schmidt, K. K. Baldridge, J. A. Boatz, S. T. Elbert, M. S. Gordon, J. H. Jensen, S. Koseki, N. Matsunaga, K. A. Nguyen, S. Su, et al. General atomic and molecular electronic structure system. *J. Comput. Chem.*, **14**, 1347 (1993).

[293] C. Schön, K. Hammerer, M. M. Wolf, J. I. Cirac, and E. Solano. Sequential generation of matrix-product states in cavity QED. *Phys. Rev. A*, **75**, 032311 (2007).

[294] C. Schön, E. Solano, F. Verstraete, J. I. Cirac, and M. M. Wolf. Sequential generation of entangled multiqubit states. *Phys. Rev. Lett.*, **95**, 110503 (2005).

[295] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran. Evaluating analytic gradients on quantum hardware. arXiv preprint arXiv:1811.11184 (2018).

[296] M. Schuld, A. Bocharov, K. Svore, and N. Wiebe. Circuit-centric quantum classifiers. arXiv preprint arXiv:1804.00633 (2018).

[297] M. Schuld and N. Killoran. Quantum machine learning in feature hilbert spaces. arXiv preprint arXiv:1803.07128 (2018).

[298] M. Schuld and F. Petruccione. *Supervised learning with quantum computers*. Springer, Cham, 2018.

[299] M. Schwarz, K. Temme, and F. Verstraete. Preparing projected entangled pair states on a quantum computer. *Phys. Rev. Lett.*, **108**, 110502 (2012).

[300] J. T. Seeley, M. J. Richard, and P. J. Love. The bravyi-kitaev transformation for quantum computation of electronic structure. *J. Chem. Phys.*, **137**, 224109 (2012).

[301] D. Sénéchal. An introduction to quantum cluster methods. arXiv preprint arXiv:0806.2690 (2008).

[302] D. Sénéchal. The variational cluster approximation for Hubbard models: Practical implementation. In *International Symposium on High Performance Computing Systems and Applications*, pages 9–15, 2008.

[303] D. Sénéchal, A.-M. Tremblay, and C. Bourbonnais. *Theoretical methods for strongly correlated electrons*. CRM Series in Mathematical Physics. Springer-Verlag New York, 2004.

[304] Y. Shen, X. Zhang, S. Zhang, J.-N. Zhang, M.-H. Yung, and K. Kim. Quantum implementation of the unitary coupled cluster for simulating molecular electronic structure. *Phys. Rev. A*, **95**, 020501 (2017).

[305] V. V. Shende, S. S. Bullock, and I. L. Markov. Synthesis of quantum-logic circuits. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, **25**, 1000 (2006).

[306] C. D. Sherrill and H. F. Schaefer. The configuration interaction method: Advances in highly correlated approaches. *Adv. Quantum Chem.*, **34**, 143 (1999).

[307] T. Shi, E. Demler, and J. I. Cirac. Variational study of fermionic and bosonic systems with non-gaussian states: Theory and applications. *Ann. Phys.*, **390**, 245 (2018).

[308] P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, **35**, 124 (1994).

[309] P. W. Shor. Introduction to quantum algorithms. In *Proc. Sympos. Appl. Math.*, volume 58, pages 143–160, 2002.

[310] A. Signoracci, T. Duguet, G. Hagen, and G. R. Jansen. Ab initio bogoliubov coupled cluster theory for open-shell nuclei. *Phys. Rev. C*, **91**, 064320 (2015).

[311] S. Sim, Y. Cao, J. Romero, P. D. Johnson, and A. Aspuru-Guzik. A framework for algorithm deployment on cloud-based quantum computers. arXiv preprint arXiv:1810.10576 (2018).

[312] H. Situ, Z. He, L. Li, and S. Zheng. Adversarial training of quantum born machine. arXiv preprint arXiv:1807.01235 (2018).

[313] R. S. Smith, M. J. Curtis, and W. J. Zeng. A practical quantum instruction set architecture. arXiv preprint arXiv:1608.03355 (2017).

[314] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25*, pages 2951–2959, 2012.

[315] A. N. Soklakov and R. Schack. Efficient state preparation for a register of quantum bits. *Phys. Rev. A*, **73**, 012307 (2006).

[316] R. Somma, G. Ortiz, J. Gubernatis, E. Knill, and R. Laflamme. Simulating physical phenomena by quantum networks. *Phys. Rev. A*, **65**, 042323 (2002).

[317] A. Sørensen and K. Mølmer. Quantum computation with ions in thermal motion. *Phys. Rev. Lett.*, **82**, 1971 (1999).

[318] A. Sørensen and K. Mølmer. Entanglement and quantum computation with ions in thermal motion. *Phys. Rev. A*, **62**, 022311 (2000).

[319] P. B. M. Sousa and R. V. Ramos. Universal quantum circuit for n-qubit quantum gate: A programmable quantum gate. arXiv preprint quant-ph/0602174 (2006).

[320] J. C. Spall. A one-measurement form of simultaneous perturbation stochastic approximation. *Automatica*, **33**, 109 (1997).

[321] J. C. Spall. Adaptive stochastic approximation by the simultaneous perturbation method. *IEEE Trans. Autom. Control*, **45**, 1839 (2000).

[322] D. S. Steiger, T. Häner, and M. Troyer. ProjectQ: an open source software framework for quantum computing. *Quantum*, **2**, 49 (2018).

[323] C. J. Stein and M. Reiher. Automated selection of active orbital spaces. *J. Chem. Theory Comput.*, **12**, 1760 (2016). PMID: 26959891.

[324] M. Steudtner and S. Wehner. Fermion-to-qubit mappings with varying resource requirements for quantum simulation. *New J. Phys.*, **20**, 063010 (2018).

[325] L. Z. Stolarczyk and H. J. Monkhorst. Quasiparticle fock-space coupled-cluster theory. *Mol. Phys.*, **108**, 3067 (2010).

[326] M. Streif and M. Leib. Comparison of qaoa with quantum and simulated annealing. arXiv preprint arXiv:1901.01903 (2019).

[327] A. Szabo and N. S. Ostlund. *Modern quantum chemistry: Introduction to advanced electronic structure theory*. McGraw-Hill / Dover Publications, 1989.

[328] P. G. Szalay, T. Müller, G. Gidofalvi, H. Lischka, and R. Shepard. Multiconfiguration self-consistent field and multireference configuration interaction methods and applications. *Chem. Rev.*, **112**, 108 (2012). PMID: 22204633.

[329] S. Taghavi, R. L. Kosut, and D. A. Lidar. Channel-optimized quantum error correction. *IEEE Trans. Inf. Theory*, **56**, 1461 (2010).

[330] A. G. Taube and R. J. Bartlett. New perspectives on unitary coupled-cluster theory. *Int. J. Quantum Chem.*, **106**, 3393 (2006).

[331] K. Temme, S. Bravyi, and J. M. Gambetta. Error mitigation for short-depth quantum circuits. *Phys. Rev. Lett.*, **119**, 180509 (2017).

[332] B. M. Terhal. Quantum error correction for quantum memories. *Rev. Mod. Phys.*, **87**, 307 (2015).

[333] B. M. Terhal and D. P. DiVincenzo. Classical simulation of noninteracting-fermion quantum circuits. *Phys. Rev. A*, **65**, 032325 (2002).

[334] B. Toloui and P. J. Love. Quantum algorithms for quantum chemistry based on the sparsity of the ci-matrix. arXiv:1312.2579 (2013).

[335] Y. Tomita and K. M. Svore. Low-distance surface codes under realistic quantum noise. *Phys. Rev. A*, **90**, 062320 (2014).

[336] A. Tranter, S. Sofia, J. Seeley, M. Kaicher, J. McClean, R. Babbush, P. V. Coveney, F. Mintert, F. Wilhelm, and P. J. Love. The bravyi-kitaev transformation: Properties and applications. *Int. J. Quantum Chem.*, **115**, 1431 (2015).

[337] A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. *P. Lond. Math. Soc.*, **2**, 230 (1937).

[338] J. Valatin. Generalized Hartree-Fock method. *Phys. Rev.*, **122**, 1012 (1961).

[339] L. G. Valiant. Quantum circuits that can be simulated classically in polynomial time. *SIAM J. Comput.*, **31**, 1229 (2002).

[340] L. Veis, A. Antalík, J. Brabec, F. Neese, Ö. Legeza, and J. Pittner. Coupled cluster method with single and double excitations tailored by matrix product state wave functions. *J. Phys. Chem. Lett.*, **7**, 4072 (2016). PMID: 27682626.

[341] L. Veis, J. Višňák, T. Fleig, S. Knecht, T. Saue, L. Visscher, and J. c. v. Pittner. Relativistic quantum chemistry on quantum computers. *Phys. Rev. A*, **85**, 030304 (2012).

[342] M. Veldhorst, C. H. Yang, J. C. C. Hwang, W. Huang, J. P. Dehollain, J. T. Muhonen, S. Simmons, A. Laucht, F. E. Hudson, K. M. Itoh, et al. A two-qubit logic gate in silicon. *Nature*, **526**, 410 (2015).

[343] G. Verdon, M. Broughton, and J. Biamonte. A quantum algorithm to train neural networks using low-depth circuits. arXiv preprint arXiv:1712.05304v1 (2017).

[344] F. Verstraete, J. I. Cirac, and J. I. Latorre. Quantum circuits for strongly correlated quantum systems. *Phys. Rev. A*, **79**, 032316 (2009).

[345] V. Veryazov, P. Å. Malmqvist, and B. O. Roos. How to select active space for multiconfigurational quantum chemistry? *Int. J. Quantum Chem.*, **111**, 3329 (2011).

[346] K. E. W. Vugrin. *On the effects of noise on parameter identification optimization problems*. PhD thesis, Virginia Tech, 2005.

[347] D. J. Wales and J. P. K. Doye. Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *J. Phys. Chem. A*, **101**, 5111 (1997).

[348] J. J. Wallman and S. T. Flammia. Randomized benchmarking with confidence. *New J. Phys.*, **16**, 103032 (2014).

[349] I. A. Walmsley and M. G. Raymer. Toward quantum-information processing with photons. *Science*, **307**, 1733 (2005).

[350] K. H. Wan, O. Dahlsten, H. Kristjánsson, R. Gardner, and M. S. Kim. Quantum generalisation of feedforward neural networks. *npj Quantum Inf.*, **3**, 36 (2017).

[351] H. Wang, S. Ashhab, and F. Nori. Efficient quantum algorithm for preparing molecular-system-like states on a quantum computer. *Phys. Rev. A*, **79**, 042335 (2009).

[352] J. Wang, S. Paesani, R. Santagati, S. Knauer, A. A. Gentile, N. Wiebe, M. Petruzzella, J. L. O

Brien, J. G. Rarity, A. Laing, et al. Experimental quantum hamiltonian learning. *Nat. Phys.*, **13**, 551 (2017).

[353] D. Wecker, M. B. Hastings, and M. Troyer. Progress towards practical quantum variational algorithms. *Phys. Rev. A*, **92**, 042303 (2015).

[354] D. Wecker, M. B. Hastings, and M. Troyer. Training a quantum optimizer. *Phys. Rev. A.*, **94**, 022309 (2016).

[355] D. Wecker, M. B. Hastings, N. Wiebe, B. K. Clark, C. Nayak, and M. Troyer. Solving strongly correlated electron models on a quantum computer. *Phys. Rev. A*, **92**, 062318 (2015).

[356] G. Wendin. Quantum information processing with superconducting circuits: a review. *Rep. Prog. Phys.*, **80**, 106001 (2017).

[357] R. White and T. Downs. *How computers work*. Que Corp., 2007.

[358] J. Whitfield. *At the intersection of quantum computing and quantum chemistry*. PhD thesis, Faculty of Arts and Sciences, 2011.

[359] J. D. Whitfield, J. Biamonte, and A. Aspuru-Guzik. Simulation of electronic structure Hamiltonians using quantum computers. *Mol. Phys.*, **109**, 735 (2011).

[360] D. W. Whitman and B. K. Carpenter. Limits on the activation parameters for automerization of cyclobutadiene-1,2-d2. *J. Am. Chem. Soc.*, **104**, 6473 (1982).

[361] M. M. Wilde. *Quantum information theory*. Cambridge University Press, 2013.

[362] D. Willsch, M. Nocon, F. Jin, H. De Raedt, and K. Michielsen. Gate-error analysis in simulations of quantum computers with transmon qubits. *Phys. Rev. A*, **96**, 062302 (2017).

[363] C. Wilson, J. Otterbach, N. Tezak, R. Smith, G. Crooks, and M. da Silva. Quantum kitchen sinks: An algorithm for machine learning on near-term quantum computers. arXiv preprint arXiv:1806.08321 (2018).

[364] D. J. Wineland. Nobel lecture: Superposition, entanglement, and raising schrödinger's cat. *Rev. Mod. Phys.*, **85**, 1103 (2013).

[365] T. Yamamoto, Y. A. Pashkin, O. Astafiev, Y. Nakamura, and J.-S. Tsai. Demonstration of conditional gate operation using superconducting charge qubits. *Nature*, **425**, 941 (2003).

[366] T. Yamazaki, S. Matsuura, A. Narimani, A. Saidmuradov, and A. Zaribafiyan. Towards the practical application of near-term quantum computers in quantum chemistry simulations: A problem decomposition approach. arXiv preprint arXiv:1806.01305 (2018).

[367] Z.-C. Yang, A. Rahmani, A. Shabani, H. Neven, and C. Chamon. Optimizing variational quantum algorithms using pontryagin's minimum principle. *Phys. Rev. X*, **7**, 021027 (2017).

[368] X. Yuan, S. Endo, Q. Zhao, S. Benjamin, and Y. Li. Theory of variational quantum simulation. arXiv preprint arXiv:1812.08767 (2018).

[369] M.-H. Yung, J. Casanova, A. Mezzacapo, J. McClean, L. Lamata, A. Aspuru-Guzik, and E. Solano. From transistor to trapped-ion computers for quantum chemistry. *Sci. Rep.*, **4**, 3589 (2014).

[370] M.-H. Yung, J. D. Whitfield, S. Boixo, D. G. Tempel, and A. Aspuru-Guzik. *Introduction to quantum algorithms for physics and chemistry*, pages 67–106. John Wiley & Sons, Inc., 2014.

[371] P. Zanardi. Virtual quantum subsystems. *Phys. Rev. Lett.*, **87**, 077901 (2001).

[372] J. Zeng, Y. Wu, J.-G. Liu, L. Wang, and J. Hu. Learning and inference on generative adversarial quantum circuits. arXiv preprint arXiv:1808.03425 (2018).

[373] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin. Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. arXiv preprint arXiv:1812.01041 (2018).

[374] G. Zhu, Y. Subaşı, J. D. Whitfield, and M. Hafezi. Hardware-efficient fermionic simulation with a cavity–QED system. *npj Quantum Inf.*, **4**, 16 (2018).

[375] E. Zohar and M. Burrello. Formulation of lattice gauge theories for quantum simulations. *Phys. Rev. D*, **91**, 054506 (2015).