UNIVERSITAT DE BARCELONA

Variational quantum architectures

Applications for noisy intermediate-scale quantum computers

Carlos Bravo Prieto

UNIVERSITAT DE BARCELONA

TESI DOCTORAL

Carlos Bravo Prieto

Variational quantum architectures

Applications for noisy intermediate-scale quantum computers

Departament de Física Quàntica i Astrofísica

Director de tesi: Dr. José Ignacio Latorre

Barcelona, Abril de 2022

Variational quantum architectures

Applications for noisy intermediate-scale quantum computers

Memòria presentada per optar al grau de doctor per la

Universitat de Barcelona

Programa de doctorat en Física

Autor: Carlos Bravo Prieto

Director: José Ignacio Latorre Sentís

Tutor: Joan Soto Riera

IMPRINT

Variational quantum architectures: applications for noisy intermediate-scale quantum computers

Printed in Spain.

COLOPHON

This thesis was typeset using LaTeX and the `memoir` documentclass. It is based on Aaron Turon's thesis *Understanding and expressing scalable concurrency*[1], itself a mixture of `classicthesis`[2] by André Miede and `tufte-latex`[3], based on Edward Tufte's *Beautiful Evidence*.

The bibliography was processed by Biblatex. The body text is set 10/14pt (long primer) on a 26pc measure. The margin text is set 8/9pt (brevier) on a 12pc measure. Matthew Carter's Charter acts as both the text and display typeface. Monospaced text uses Jim Lyles's `Bitstream Vera Mono` ("Bera Mono").

[1] `https://people.mpi-sws.org/~turon/turon-thesis.pdf`

[2] `https://bitbucket.org/amiede/classicthesis/`

[3] `https://github.com/Tufte-LaTeX/tufte-latex`

Agradecimientos

Primero de todo, me gustaría dar las gracias a José Ignacio Latorre, supervisor de mi tesis doctoral. Le agradezco todas las horas invertidas discutiendo sobre tantos y variados temas. Para mí ha sido un modelo a seguir, especialmente por su perseverancia, inteligencia y empatía. Su enorme experiencia y valores me han ayudado sin duda a crecer; espero ser capaz de transmitir lo mismo en el futuro. Estaré siempre en deuda con él por haberme dado la oportunidad de recorrer este camino.

During my Ph.D. I had the opportunity to work in many different places around the globe. I want to thank Lukasz Cincio and Patrick J. Coles for giving me the opportunity to work at Los Alamos National Laboratory for a summer internship. I am also grateful for their mentorship and for teaching me that hard work always pays off.

I am thankful to all the people in Barcelona for their friendship and fruitful conversations: Jorge Cortada, Pol Forn, Artur Garcia, Elies Gil, David López, Josep Lumbreras, Jordi Planagumà, Luca Tagliacozzo, and Chris Warren. Special thanks to Alba Cervera, Diego García, Adrián Pérez, and Sergi Ramos, with whom at some point I shared the sorrows and glories of doing a Ph.D.

After my first two years of Ph.D. in Barcelona, I moved to Abu Dhabi for another two years to continue my research at the Technology Innovation Institute. I was lucky to find that it was full of brilliant researchers who now I can consider colleagues, including (but not limited to) Luigi Amico, Leandro Aolita, Ben Blain, Giancarlo Camilo, Wayne Chetcuti, Stavros Efthymiou, Thais de Lima, Ruge Lin, Marc Manzano, Giampiero Marchegiani, Claudia Núñez, Álvaro Orgaz, Juan Polo, Ingo Roth, and Javier Serrano. Thanks for the fun times and for sharing your knowledge.

I am grateful to Leandro Aolita, Stefano Carraza, Artur Garcia, and Luca Tagliacozzo for mentoring and guiding me at different stages of my Ph.D.

Furthermore, I want to acknowledge my collaborators for their work, inspiring discussions, and great ideas: Najwa Aaraj, Luigi Amico, Tony J. G. Apollaro, Julien Baglio, Emanuele Bellini, Stefano Carrazza, Marco Cè, Marco Cerezo, Wayne Chetcuti, Lukasz Cincio, Patrick J. Coles, Mirko Consiglio, Stavros Efthymiou, Andre Esser, Anthony Francis, Diego García, Dorota M. Grabowska, Ryan LaRose, Ruge Lin, Josep Lumbreras, Marc Manzano, Adrián Pérez, Sergi Ramos, Yiğit Subaşı, and Luca Tagliacozzo.

Gracias a todos los amigos de fuera de la investigación con los que he podido compartir momentos de felicidad y desconexión: Adrià Bonet, Xavi Casals, Robert Fonoll, Pere Gironella, Jorge Molina, Carles Moreno, Laura Rabanal, Albert Rodríguez, Joshua Rodríguez, Oscar Romero, Samuel Rosende, Carlos Sánchez, Juan Telechea, Isaac del Toro y David Vázquez.

No quisiera olvidarme de quién posiblemente fue mi primera inspiración para hoy estar aquí; gracias Adolf Cortel por ser el primero que confió en mí, y por enseñarme la belleza de la física.

Me gustaría también dar las gracias a mi familia, en especial a mis padres, Juan Carlos Bravo y Antonia Prieto, y a mi hermana, Laida Bravo, por todo el cariño y apoyo incondicional recibido. Por último, quiero dar las gracias a Anastasia Nikulina, la cual ha sido una pieza fundamental para que yo haya podido completar esta tesis. Sin ti no hubiera sido posible.

Author publications

Publications sorted in chronological order.

► PUBLICATIONS INCLUDED IN THIS THESIS

- C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subaşı, L. Cincio, and P. J. Coles, *Variational quantum linear solver*, arXiv preprint arXiv:1909.05820 (2019).
- C. Bravo-Prieto, J. Lumbreras-Zarapico, L. Tagliacozzo, and J. I. Latorre, *Scaling of variational quantum circuit depth for condensed matter systems*, Quantum **4**, 272 (2020).
- C. Bravo-Prieto, D. García-Martín, and J. I. Latorre, *Quantum singular value decomposer*, Physical Review A **101**, 062310 (2020).
- C. Bravo-Prieto, *Quantum autoencoders with enhanced data encoding*, Machine Learning: Science and Technology **2**, 035028 (2021).
- S. Efthymiou, S. Ramos-Calderer, C. Bravo-Prieto, A. Pérez-Salinas, D. García-Martín, A. Garcia-Saez, J. I. Latorre, and S. Carrazza, *Qibo: a framework for quantum simulation with hardware acceleration*, Quantum Science and Technology **7**, 015018 (2021).
- C. Bravo-Prieto, J. Baglio, M. Cè, A. Francis, D. M. Grabowska, and S. Carrazza, *Style-based quantum generative adversarial networks for Monte Carlo events*, arXiv preprint arXiv:2110.06933 (2021).

► OTHER PUBLICATIONS

- A. Pérez-Salinas, D. García-Martín, C. Bravo-Prieto, and J. I. Latorre, *Measuring the tangle of three-qubit states*, Entropy **22**, 436 (2020).
- S. Ramos-Calderer, A. Pérez-Salinas, D. García-Martín, C. Bravo-Prieto, J. Cortada, J. Planagumà, and J. I. Latorre, *Quantum unary approach to option pricing*, Physical Review A **103**, 032414 (2021).
- M. Consiglio, W. J. Chetcuti, C. Bravo-Prieto, S. Ramos-Calderer, A. Minguzzi, J. I. Latorre, L. Amico, and T. J. G. Apollaro, *Variational quantum eigensolver for SU(N) fermions*, arXiv preprint arXiv:2106.15552 (2021).
- S. Ramos-Calderer, C. Bravo-Prieto, R. Lin, E. Bellini, M. Manzano, N. Aaraj, and J. I. Latorre, *Solving systems of Boolean multivariate equations with quantum annealing*, Physical Review Research **4**, 013096 (2022).
- Andrea Delgado, et al., *Quantum Computing for Data Analysis in High-Energy Physics*, arXiv preprint arXiv:2203.08805 (2022).

Abstract

Quantum algorithms showing promising speedups with respect to their classical counterparts already exist. However, noise limits the quantum circuit depth, making the practical implementation of many such quantum algorithms impossible nowadays. In this sense, variational quantum algorithms offer a new approach, reducing the requisites of quantum computational resources at the expense of classical optimization. Disciplines in which variational quantum algorithms may have practical applications include simulation of quantum systems, solving large systems of linear equations, combinatorial optimization, data compression, quantum state diagonalization, among others.

This thesis studies different variational quantum algorithm applications. In Chapter 1, we introduce the main building blocks of variational quantum algorithms. In Chapter 2, we benchmark the seminal variational quantum eigensolver algorithm for condensed matter systems. In Chapter 3, we explore how the task of compressing quantum information is affected by data encoding in variational quantum circuits. In Chapter 4, we propose a novel variational quantum algorithm to compute the singular values of pure bipartite states. In Chapter 5, we develop a new variational quantum algorithm to solve linear systems of equations. Finally, in Chapter 6, we implement quantum generative adversarial networks for generative modeling tasks. The conclusions of this thesis are exposed in Chapter 7. Furthermore, supplementary material can be found in the appendices. Appendix A provides an introduction to `Qibo`, a framework for quantum simulation. Appendix B presents some results related to the Solovay-Kitaev theorem. Extra results from Chapter 5 and Chapter 6 can be found in Appendix C and Appendix D, respectively.

Contents

AGRADECIMIENTOS v

AUTHOR PUBLICATIONS vii

ABSTRACT ix

I INTRODUCTION 1

1 VARIATIONAL QUANTUM ALGORITHMS 3
- 1.1 Ansatz for state preparation 5
- 1.2 Characterizing the problem with the cost function 7
- 1.3 Parameter optimization 8

II APPLICATIONS OF VARIATIONAL QUANTUM ALGORITHMS 11

2 SCALING OF VARIATIONAL QUANTUM CIRCUIT DEPTH FOR CONDENSED MATTER SYSTEMS 13
- 2.1 Variational quantum eigensolver 14
- 2.2 Numerical characterization 15
  - 2.2.1 The models 16
  - 2.2.2 Gapped Hamiltonians, the perturbative regime 17
  - 2.2.3 Scaling of the accuracy at criticality, the two regimes 19
  - 2.2.4 Scaling of the entanglement entropy at criticality 21
- 2.3 Outlook 23

3 DATA ENCODING IN QUANTUM AUTOENCODERS 25
- 3.1 Quantum autoencoders 25
- 3.2 Enhanced feature quantum autoencoder 27
  - 3.2.1 Overview 27
  - 3.2.2 Cost function 28
  - 3.2.3 Ansatz 28
- 3.3 Numerical simulation: 1D Ising spin chain 30
- 3.4 Outlook 32

4 QUANTUM SINGULAR VALUE DECOMPOSITION 35
- 4.1 Classical singular value decomposition 36
- 4.2 Circuit for the quantum singular value decomposition 37
- 4.3 Variational quantum singular value decomposer 39
- 4.4 Additional applications 41
  - 4.4.1 SWAP with classical communication 41
  - 4.4.2 Quantum singular value decomposer as an encoder 42
- 4.5 Outlook 43

5 SOLVING LINEAR SYSTEMS OF EQUATIONS 45
- 5.1 Variational quantum linear solver 46
  - 5.1.1 Cost functions 47
  - 5.1.2 Operational meaning of cost functions 49
  - 5.1.3 Cost evaluation 50

5.1.4 Ansatz 52
5.1.5 Training algorithm 54
5.2 Heuristic scaling 55
5.2.1 Ising-inspired quantum linear system problem 55
5.2.2 Randomly-generated quantum linear system problem . 57
5.3 Implementation on quantum hardware 58
5.4 Outlook 59

III QUANTUM GENERATIVE MODELING 61

6 QUANTUM GENERATIVE ADVERSARIAL NETWORKS FOR MONTE CARLO EVENTS 63
6.1 Generative adversarial learning 64
6.2 Optimization procedure 66
6.3 Style-based quantum generator ansatz 66
6.4 Validation examples 68
6.4.1 1D Gamma distribution 68
6.4.2 3D correlated Gaussian distribution 70
6.5 Generating LHC events 72
6.6 Sampling from quantum hardware 74
6.7 Outlook 77

IV CONCLUSION 79

7 FINAL REMARKS 81

BIBLIOGRAPHY 83

V APPENDIX 95

A QIBO: A FRAMEWORK FOR QUANTUM SIMULATION 97

B SOLOVAY-KITAEV THEOREM 101

C SMALL SCALE IMPLEMENTATIONS OF THE LINEAR SYSTEM PROBLEM 105

D GENERATIVE MODELING WITH A NOISY SIMULATION 113

Part I

INTRODUCTION

1

Variational quantum algorithms

> *"A classical computation is like a solo voice; one line of pure tones succeeding each other. A quantum computation is like a symphony; many lines of tones interfering with one another."*
> —Seth Lloyd

Quantum computing is a new paradigm whereby quantum phenomena are harnessed to perform computations. Quantum computers are expected to be able to perform computational tasks with a broad range of applications, such as quantum simulation [BN09; GAN14], search and optimization [Gro97; Bra+02], or tackle math-related problems [Sho99; Kup05; Hal07; HHL09]. Most of the aforementioned quantum algorithms are likely to outperform any existing classical algorithm [San08; CVD10; Mon16]. However, they cannot be implemented on current quantum computers without quantum error correction. Reducing errors on physical qubits beyond the required fault tolerance is a cornerstone in experimental quantum computing, yet scaling these systems up to the level required for large-scale computations, at the present day, is still far from current experimental capabilities [Fow+12; Bar+14].

Large-scale fault-tolerant quantum computation is, therefore, a rather distant dream, typically estimated to be at least two decades ahead. A reasonable question then is whether we can do something useful with the existing noisy intermediate-scale quantum (NISQ) devices [Pre18]. In such near-term devices, *i.e.*, of a few tens or hundreds of qubits, noise limits the problem size to be considerably small. An emerging paradigm to make use of NISQ computers is to implement variational quantum algorithms (VQAs) [End+21; Cer+21a], which manage to reduce the requisites of quantum computational resources, that is, circuit depth, coherence, and connectivity between the qubits, at the expense of classical computation. In particular, these algorithms leverage the power of classical optimization methods while still employing quantum computers for, ideally, speedups. The action of the algorithm is characterized by a shallow sequence of parameterized quantum gates with a trainable set of classical parameters and then an optimization procedure to minimize a cost function with machine-learning techniques.

Due to the optimization procedure, all VQAs are heuristic algorithms, making rigorous complexity analysis difficult. In addition, one is not guaranteed to reach the optimal solution due to the existence of local minima, and we can not ensure that the procedure yields better approximations to the solution than current classical methods. Serious efforts are put into this field to obtain new theoretical and experimental ideas and practical applications.

[BN09] Buluta and Nori, "Quantum simulators"

[GAN14] Georgescu et al., "Quantum simulation"

[Gro97] Grover, "Quantum mechanics helps in searching for a needle in a haystack"

[Bra+02] Brassard et al., "Quantum amplitude amplification and estimation"

[Sho99] Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer"

[Kup05] Kuperberg, "A subexponential-time quantum algorithm for the dihedral hidden subgroup problem"

[Hal07] Hallgren, "Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem"

[HHL09] Harrow et al., "Quantum algorithm for linear systems of equations"

[Pre18] Preskill, "Quantum Computing in the NISQ era and beyond"

[End+21] Endo et al., "Hybrid quantum-classical algorithms and quantum error mitigation"

[Cer+21a] Cerezo et al., "Variational quantum algorithms"

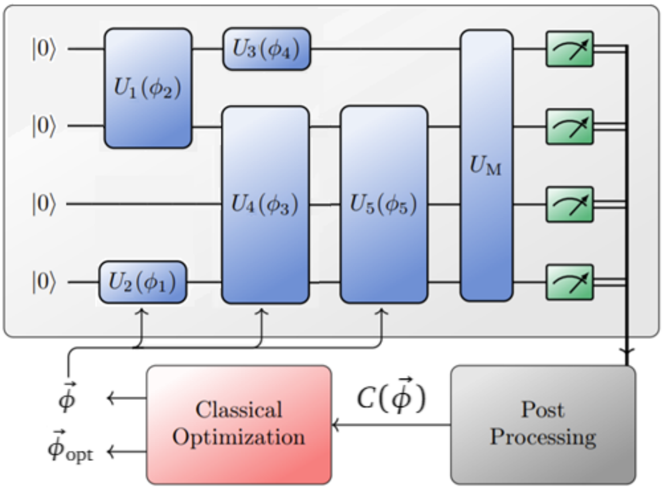

FIGURE 1.1: General schematic diagram of a VQA. The input of a VQA is a quantum circuit U parameterized by a set of classical parameters $\vec{\phi}$, and a cost function C that characterizes the problem we aim to solve. The task of the QPU is to prepare a quantum state $U(\vec{\phi})|0\rangle = |\varphi(\vec{\phi})\rangle$. Once this trial quantum state has been prepared, measurements are performed in order to evaluate the cost function $C(\vec{\phi})$. The result of this evaluation is fed into the CPU, where parameters $\vec{\phi}$ in the ansatz $U(\vec{\phi})$ are adjusted. This quantum-classical optimization loop is repeated until the cost $C(\vec{\phi})$ is below a user-specified threshold. When this loop terminates and the optimal parameters $\vec{\phi}_{\text{opt}}$ are found, the resulting quantum circuit $U(\vec{\phi}_{\text{opt}})$ prepares the solution of the problem. Figure adapted from Ref. [BK21].

A variational quantum algorithm is a hybrid approach in which a quantum computer, sometimes referred to as Quantum Processing Unit (QPU), cooperates with a Classical Processing Unit (CPU)[1] to solve a particular problem. A graphical depiction of the general scheme followed by VQAs is shown in Figure 1.1. It can be broken down into the following steps:

- The task of the QPU is to prepare a trial state wavefunction $|\varphi(\vec{\phi})\rangle$ by applying a unitary quantum circuit $U(\vec{\phi})$ that depends upon a set of classical parameters $\vec{\phi}$, typically characterizing angles of single-qubit rotations or parameterized two-qubit gates.
- Once the trial state $|\varphi(\vec{\phi})\rangle$ has been prepared, measurements are performed on one or more qubits. This process is repeated in order to accumulate statistics and evaluate a classical cost function $C(\vec{\phi})$ defined in terms of quantum observables (*i.e.* hermitian operators). This evaluation may involve some classical post-processing, such as *e.g.* computing a weighted sum of mean values for observables coming out of the QPU.
- The result of this evaluation is fed into the CPU, which then uses some classical minimization routine to adjust the circuit parameters $\vec{\phi}$ and return a new set of values, which are employed in the preparation of the next trial state. This step may involve new calls to the QPU, for instance, to compute quantum gradients.
- This quantum-classical loop is repeated until user-specified convergence criteria for the minimization of the cost function are fulfilled. Once this loop is terminated and the optimal parameters $\vec{\phi}_{\text{opt}}$ are found, the resulting quantum circuit $U(\vec{\phi}_{\text{opt}})$ prepares the solution to the problem, $|\varphi(\vec{\phi}_{opt})\rangle$.

[1] Notice that, in general, the CPU may be composed of several central processing units, combined with dedicated Graphical Processing Units (GPUs) or Tensor Processor Units (TPUs).

The procedure sketched above is the basic common strategy for all VQAs. Different VQAs that solve different problems will differ in the architecture of the quantum circuit they apply or in the cost function definition. Indeed, the VQA framework provides the tools to tackle a wide variety of problems [Bha+22], and in fact, it has been shown that VQAs admit a universal model of quantum computation [Bia21].

[Bha+22] Bharti et al., "Noisy intermediate-scale quantum algorithms"

[Bia21] Biamonte, "Universal variational quantum computation"

Several important topics deserve consideration concerning VQAs, and we will treat them separately in the following sections. In the first place, one has to define a particular structure for the parameterized quantum circuit, often referred to as ansatz. Then, one also has to define a cost function whose minimization below a certain threshold solves the problem. As we shall see, this cost function needs to meet specific requirements to be of practical use. Lastly, one has to choose an optimization strategy, and we shall see that the exponential size of the Hilbert space and the problem of vanishing gradients directly impact the choice of all these three components of VQAs.

1.1 ANSATZ FOR STATE PREPARATION

One of the main ideas behind VQAs is to retain the advantage of preparing classically inaccessible states despite the low coherence times. Indeed, the parameterized quantum circuits provide access to a manifold of quantum states over an exponentially large Hilbert space. Consequently, the flexibility of the VQA to approximate the solution heavily depends on the choice of ansatz. In general, constructing an arbitrary n-qubit quantum state requires exponential resources in terms of gates and classical parameters. The complete rationale of this argument is just a matter of counting degrees of freedom. Regardless, we usually deal with problems with a strong relation with physical systems, where the number of classical parameters that characterize such quantum states may be polynomially large instead of exponential. This is because physical Hamiltonians are usually described by local interactions, which means that each qubit mostly communicates with other nearby qubits.

We thus are in the framework of finding the best approximation $U(\vec{\phi})$ to a unitary transformation V given an elementary set of quantum gates, being $V\left|0\right\rangle^{\otimes n}$ the solution of the problem. This framework has analytical bounds on the error induced by approximating a unitary transformation with a finite number of elementary gates. The Solovay-Kitaev theorem states that an arbitrary unitary acting on n qubits can be approximated with an error ε by using at most $\mathcal{O}(\log^c(1/\varepsilon))$ elementary gates chosen appropriately from a universal set of quantum gates closed under inversion, where $c \sim 4$ [DN06; NC11]. Alternative versions of the theorem have lowered the value of c [Kit+02], however there is an optimal value of $c = 1$ [HRC02; NC11]. This theorem suggests that if the ansatz of our VQA is built from a set of universal gates, then by increasing the number of gates, we will be able to arbitrarily reduce the error between our approximation and the solution of the problem.

[Kit+02] Kitaev et al., *Classical and Quantum Computation*

[HRC02] Harrow et al., "Efficient discrete approximations of quantum gates"

[NC11] Nielsen and Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*

From the practical point of view of building the variational quantum circuit $U(\vec{\phi})$ that best approximates V from a given number of gates, the

Solovay-Kitaev theorem is not very useful. For example, it does not specify on which constituents each of the elementary gates should act and how elementary gates should be concatenated. Indeed, there is no general recipe for constructing an ansatz for a VQA that will universally work on every problem. Yet, we may classify different designs of variational circuits in two main categories, namely, hardware efficient ansatz and physically-motivated ansatz.

- Hardware efficient ansatz. At the time of writing, this is the most commonly used ansatz architecture. The original motivation of this type of parameterized circuit is to overcome the existing limitations of the quantum hardware [Kan+17] by designing the ansatz so it can be adapted to the chip architecture and accommodate the quantum gates for maximizing the fidelity of the output state. The general construction of a hardware efficient ansatz comprises L entangling blocks (or layers) acting on the qubits, consisting of parameterized single-qubit rotations and entangling gates, usually given by

[Kan+17] Kandala et al., "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets"

$$U(\vec{\phi}) = \sum_{k=1}^{L} U_k(\phi_k) W_k \, , \tag{1.1}$$

where U_k are the parameterized unitaries, and W_k represent non-parameterized quantum gates.

The main advantages of the hardware efficient ansatz, that is, entanglement production, expressibility, and accuracy, are studied in Refs. [SJAG19; Woi+20; BP+20]. It also has been shown this ansatz can accommodate problem symmetries to improve the optimization convergence [Bar+18; Gan+19; Gar+20; Bar+21]. Although the hardware efficient ansatz is very suitable to be implemented in current NISQ computers, this ansatz has no structure. It can lead to trainability issues when randomly initialized, and therefore, is not expected to work that well for large systems, as we will discuss in Sec. 1.3.

[SJAG19] Sim et al., "Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms"

[Woi+20] Woitzik et al., "Entanglement production and convergence properties of the variational quantum eigensolver"

[BP+20] Bravo-Prieto et al., "Scaling of variational quantum circuit depth for condensed matter systems"

[Bar+18] Barkoutsos et al., "Quantum algorithms for electronic structure calculations: Particle-hole Hamiltonian and optimized wave-function expansions"

[Gan+19] Ganzhorn et al., "Gate-efficient simulation of molecular eigenstates on a quantum computer"

[Gar+20] Gard et al., "Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm"

[Bar+21] Barron et al., "Preserving symmetries for variational quantum eigensolvers in the presence of noise"

- Physically-Motivated Ansatz. In general, an arbitrary unitary operation can be generated as a time evolution operator such as

$$U(t) = e^{i\hat{g}t} \, , \tag{1.2}$$

where t defines the time evolution and $\hat{g}$ is a hermitian operator, usually referred to as the generator. Physically-motivated ansatzë are constructed following time evolutions in the form of Eq. 1.2, where the generator $\hat{g}$ is derived from physical properties of the system of interest. The main example of physically-motivated ansatz is the unitary coupled-cluster ansatz [HS88; TB06], where the generators are elementary fermionic excitations, and it is mainly used for quantum chemistry applications [McC+16; Rom+18; Oll+20; Miz+20; Wan+21a].

Another prominent example is used in the Quantum Approximate Optimization Algorithm (QAOA), which was originally introduced to solve combinatorial optimization problems [FGG14]. The ansatz used in QAOA involves an alternating structure usually referred to as quantum alternating operator ansatz [Had+19]. This ansatz is inspired by a Trotterized adiabatic

[FGG14] Farhi et al., "A quantum approximate optimization algorithm" evolution [Suz76; Far+00], and reads

$$U(\vec{\lambda},\vec{\beta}) = \prod_{l=1}^{p} e^{-i\beta_l H_M} e^{-i\lambda_l H_P}\,, \tag{1.3}$$

where the order p of the Trotterization determines the precision of the solution, $\vec{\lambda}$ and $\vec{\beta}$ are the tunable parameters, and H_P and H_M are hermitian operators known as the problem Hamiltonian and the mixer Hamiltonian, respectively. The goal of this ansatz is to map an input state $|\Psi_0\rangle$, typically prepared by acting with Hadamard gates on every qubit, to the ground state of the problem Hamiltonian H_P.

Notice that the ansatzë mentioned above keep the circuit structure fixed. At the same time, the optimization is done over a continuous set of variables, typically encoded as the angle of single-qubit rotations. However, one can optimize the circuit structure itself instead, adding and removing quantum gates. This is known as variable structure ansatz, which has also shown promising performance [Gri+19; Rat+19; Bil+21; Tan+21].

[Gri+19] Grimsley et al., "An adaptive variational algorithm for exact molecular simulations on a quantum computer"

[Rat+19] Rattew et al., "A domain-agnostic, noise-resistant, hardware-efficient evolutionary variational quantum eigensolver"

[Bil+21] Bilkis et al., "A semi-agnostic ansatz with variable structure for quantum machine learning"

[Tan+21] Tang et al., "Qubit-ADAPT-VQE: An adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor"

1.2 CHARACTERIZING THE PROBLEM WITH THE COST FUNCTION

The cost function is an essential constituent of every VQA, as it completely characterizes the target problem. As a consequence, the cost function has to be carefully designed. To do so, there exist certain criteria that any such function should ideally fulfill [Kha+19b], namely:

[Kha+19b] Khatri et al., "Quantum-assisted quantum compiling"

- FAITHFULNESS. The global minimum is achieved if and only if the state $|\varphi(\vec{\phi}_{opt})\rangle$ corresponds to the exact solution to the problem.
- EFFICIENTLY COMPUTABLE. This means that the number of quantum gates and measured observables needed to evaluate the cost function has to scale at most polynomially with the system size.
- OPERATIONAL MEANING. That is, we should be able to quantify how close we are to the exact solution by looking at the value of the cost function.
- TRAINABLE. This implies that the cost function needs to be "well-behaved" as we increase the size of the problem. We discuss this in more detail in Sec. 1.3.

Every quantum computation involves measurements of observables to extract information from the quantum computer, and so do VQAs. This implies that the cost function of a VQA can always be ultimately defined in terms of the expectation value of a certain Hamiltonian, which we call problem Hamiltonian H_P. The cost function then reads $C \equiv \langle\varphi(\vec{\phi})|H_p|\varphi(\vec{\phi})\rangle$, and the solution to the problem is the ground state itself or otherwise it is encoded in it. Recall that VQAs rely upon the Rayleigh-Ritz variational principle. Namely, we know that the expectation value of the Hamiltonian is the sum of the eigenenergies E_n, each weighted by their corresponding amplitudes. When deriving the variational principle, we replace all the E_n by E_0, *i.e.* the ground state energy. This results in the following bound

$$E_0 \leq \frac{\langle\varphi(\vec{\phi})|H_p|\varphi(\vec{\phi})\rangle}{\langle\varphi(\vec{\phi})|\varphi(\vec{\phi})\rangle} \equiv C\,, \tag{1.4}$$

where $E_0 = C$ if and only if $|\varphi(\vec{\phi})\rangle$ is the ground state of the Hamiltonian H_p, and therefore the solution of the problem. Otherwise, the cost function is always larger than E_0. This, of course, does not mean that a VQA needs always be defined in Hamiltonian form, for other alternative definitions may, in some cases, be conceptually more appealing and thus help algorithm design.

A class of problems for which the cost function defined in terms of H_p appears naturally are those related to physical systems. The Hamiltonian collects all the contributions to the total energy from the different interacting particles or elements in the system. A straightforward manner of expressing this type of Hamiltonian is expanding it as a sum of Pauli operators, *i.e.* tensor products of Pauli matrices, $\{\mathbb{1}, \sigma_x, \sigma_y, \sigma_z\}$ acting on different qubits, and then measuring in the corresponding bases to estimate the expectation values of all the terms [MF19]. Recall that, in general, measurements on available quantum computers can only be done on the σ_z basis, but they can be simulated on another basis with the help of additional gates, namely an H-gate for σ_x and an $\mathrm{S}^\dagger$-gate followed by an H-gate for σ_y. However, no trivial decomposition might be obtained for a large number of qubits and arbitrary Hamiltonians. In such a case, alternative methods can be used to compute the cost function. The common way of proceeding is using the standard Hadamard test [AJL09], or Swap test [Buh+01; GC01], depending on whether one wants to explicitly obtain the expectation value of an observable or measure the overlap between two quantum states. Nevertheless, several techniques more suited to NISQ devices have been proposed recently, such as the Bell-Basis algorithm [Cin+18], the Hilbert-Schmidt test [Kha+19b], or the Hadamard-Overlap test [BP+19]. These methods avoid using as many gates in exchange of *e.g.* more qubits or classical post-processing.

[MF19] Mitarai and Fujii, "Methodology for replacing indirect measurements with direct measurements"

[AJL09] Aharonov et al., "A polynomial quantum algorithm for approximating the Jones polynomial"

[Buh+01] Buhrman et al., "Quantum fingerprinting"

[GC01] Gottesman and Chuang, "Quantum digital signatures"

[Cin+18] Cincio et al., "Learning the quantum algorithm for state overlap"

[Kha+19b] Khatri et al., "Quantum-assisted quantum compiling"

[BP+19] Bravo-Prieto et al., "Variational quantum linear solver"

Finally, it is noteworthy to mention that VQAs offer a "cost-function robustness" against coherent errors [McC+16]. Specifically, when trying to implement a unitary $U(\vec{\phi})$, it may happen that, given coherent errors in the physical implementation, the applied unitary is instead $\tilde{U}(\vec{\phi})$. The VQA may variationally suppress these errors by applying $\tilde{U}(\vec{\phi}+\vec{\alpha}) = U(\vec{\phi})$. Therefore, the optimal value of the cost function may still be found without detailed knowledge of the error mechanism.

[McC+16] McClean et al., "The theory of variational hybrid quantum-classical algorithms"

1.3 PARAMETER OPTIMIZATION

Machine-learning methods heavily rely on optimal performance of parameter optimization, and VQAs are not an exception. Indeed, their performance depends on the optimization efficiency and reliability. Notice, however, that current quantum computers are noisy, and the stochastic nature coming from the finite amount of measurements makes the optimization even harder in practice. Furthermore, it has been shown that the classical optimization of VQAs is an NP-hard problem [BK21].

[BK21] Bittel and Kliesch, "Training variational quantum algorithms is NP-hard"

Among many of the aforementioned inherent challenges, the main obstacle VQAs have to deal with is the barren plateau phenomenon [McC+18]. It has been shown that cost function gradients vanish to almost zero in every direction of the Hilbert space when using randomly initialized ansatzë, which becomes exponentially worse as we increase the circuit depth and the num-

[McC+18] McClean et al., "Barren plateaus in quantum neural network training landscapes" ber of qubits. Besides the barren plateau phenomenon introduced by choice of ansatz, other physical phenomena can generate them. For instance, noise and decoherence also generate this problem [Wan+21b]. Entanglement-induced barren plateaus have also been reported in Ref. [MKW21]. Lastly, it has been shown as well that the choice of cost function influences the appearance of barren plateaus [Cer+21b; UB21]. Although further work is needed on this topic, some interesting proposals can be found in literature, such as strategies for parameter initialization [GSL18; Gra+19; CLKAG21; Zha+22], correlation between parameters [VC21], filtering operators [Ama+22] or exploring circuit architectures [Sha+20a; Wie+20; Pes+21; Hol+22].

Let us briefly review in the following the common optimization strategies for VQAs. For convenience, we present two classes of optimization, gradient-based and gradient-free approaches.

[Wan+21b] Wang et al., "Noise-induced barren plateaus in variational quantum algorithms"

[MKW21] Marrero et al., "Entanglement-induced barren plateaus"

[Cer+21b] Cerezo et al., "Cost function dependent barren plateaus in shallow parametrized quantum circuits"

[UB21] Uvarov and Biamonte, "On barren plateaus and cost function locality in variational quantum algorithms"

- GRADIENT-BASED METHODS. Gradient-based approaches make iterative steps in the directions indicated by the gradient. Common methods imported from the classical machine learning community are `ADAM` [KB14], `ADADELTA` [Zei12], or `ADAGRAD` [DHS11], which differently adapt the size of the steps taken. Inspired by these methods, some resource-aware optimization strategies have also been proposed [Küb+20; Swe+20]. Another interesting approach is based on quantum imaginary time evolution to update the parameters [McA+19]. Similarly, another approach called quantum natural gradient descent has been proposed [Sto+20], which is based on notions of information geometry and has also been extended to incorporate noisy environments [KB19].

[KB14] Kingma and Ba, "Adam: A method for stochastic optimization"

[Zei12] Zeiler, "Adadelta: an adaptive learning rate method"

[DHS11] Duchi et al., "Adaptive subgradient methods for online learning and stochastic optimization"

[McA+19] McArdle et al., "Variational ansatz-based quantum simulation of imaginary time evolution"

[Sto+20] Stokes et al., "Quantum natural gradient"

- GRADIENT-FREE METHODS. Gradient-free approaches are those that do not directly rely on gradients to update the parameters. For instance, popular methods are based on evolutionary strategies [BS02], which are black-box optimization tools that use search distributions, and have shown promising results [Zha+20], especially when dealing with the barren plateau phenomenon [ADAG21]. Encouraged by successful results in classical machine learning, reinforcement learning techniques have also been increasingly popular for the context of VQAs [GSR19; Kha+19a; Wau+20; YBL20; HM21]. Finally, another noteworthy approach is the simultaneous perturbation stochastic approximation, or SPSA method [Spa+92]. Its main peculiarity is that it requires only two measurements of the objective function to update the parameters, and therefore, it reduces the expense of computing many gradient components at each iteration.

[BS02] Beyer and Schwefel, "Evolution strategies–a comprehensive introduction"

[Zha+20] Zhao et al., "Natural evolution strategies and variational Monte Carlo"

[ADAG21] Anand et al., "Natural evolutionary strategies for variational quantum computation"

[Spa+92] Spall et al., "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation"

Part II

APPLICATIONS OF VARIATIONAL QUANTUM ALGORITHMS

2

Scaling of variational quantum circuit depth for condensed matter systems

> *"Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy."*
> —Richard P. Feynman

Future large-scale fault-tolerant quantum computers will allow to simulate quantum systems made by a large number of constituents, thus providing important insight on their properties [BN09; BMK10; GAN14]. In particular, they will allow to characterize ground and equilibrium states of those systems through appropriately designed quantum algorithms such as those proposed in Refs. [AL99; VCL09; Tem+11; JLP12; Ber+18; Cao+19]. However, as of today, such large-scale fault-tolerant quantum computers still do not exist. Currently, noisy intermediate-scale quantum (NISQ) [Pre18] devices are already available in the labs. It seems natural then to explore how these machines can provide an understanding of the properties of quantum systems.

[BN09] Buluta and Nori, "Quantum simulators"

[BMK10] Brown et al., "Using quantum computers for quantum simulation"

[GAN14] Georgescu et al., "Quantum simulation"

[AL99] Abrams and Lloyd, "Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectors"

[VCL09] Verstraete et al., "Quantum circuits for strongly correlated quantum systems"

[Tem+11] Temme et al., "Quantum Metropolis sampling"

[JLP12] Jordan et al., "Quantum Algorithms for Quantum Field Theories"

[Ber+18] Berry et al., "Improved techniques for preparing eigenstates of fermionic Hamiltonians"

[Cao+19] Cao et al., "Quantum Chemistry in the Age of Quantum Computing"

In this chapter, we present the Variational Quantum Eigensolver (VQE) [Per+14; Til+21], a variational quantum algorithm that is designed to provide an approximation to the ground state of many-body quantum systems using NISQ devices. We benchmark the accuracy of a VQE based on a finite-depth quantum circuit encoding the ground state of local Hamiltonians. Specifically, we study condensed matter Hamiltonians. We consider a special class of quantum circuits made by several layers of unitaries that act on a pair of contiguous qubits. The unitaries are chosen from a simple set of gates. We review the reason why we use such a structure and characterize its power numerically. For gapped Hamiltonians, we can make a direct connection between our quantum circuit and perturbation theory and show how the accuracy of the ansatz increases exponentially with the number of layers. For critical systems, we observe the appearance of two regimes, one where the physics is dictated by an effective correlation length induced by the number of layers of the circuit and another one where the correlation length is actually set by the system size as expected. In particular, we discuss how the tension between the finite speed of propagation of the correlations consequence of Lieb-Robinson bounds and the growth of entanglement in critical systems is responsible of the linear scaling, with system size n, of the critical number of layers $l^*(n)$ that determines the location of the cross-over between the two regimes.

[Per+14] Peruzzo et al., "A variational eigenvalue solver on a photonic quantum processor"

[Til+21] Tilly et al., "The Variational Quantum Eigensolver: a review of methods and best practices"

2.1 VARIATIONAL QUANTUM EIGENSOLVER

Estimating low-lying eigenstates of Hamiltonians and their corresponding eigenvalues is a relevant problem in many fields of physics, ranging from condensed matter physics [Bau+12; Orú14] to classical combinatorial optimization [LK75; Tov84]. Quantum computers are not likely to efficiently solve the hardest instances of this problem since it is known to be QMA-hard [Osb12]. However, it is expected that approximate solutions of the ground state could be obtained, especially those related to physical systems.

[Bau+12] Baumgärtner et al., *The Monte Carlo method in condensed matter physics*

[Orú14] Orús, "A practical introduction to tensor networks: Matrix product states and projected entangled pair states"

[LK75] Lenstra and Kan, "Some simple applications of the travelling salesman problem"

[Tov84] Tovey, "A simplified NP-complete satisfiability problem"

[Osb12] Osborne, "Hamiltonian complexity"

In this sense, the VQE was proposed to provide a near-term solution to finding the ground state of a particular Hamiltonian H. We denote by $\tilde{U}$ the approximation of the unitary U that should rotate the initial product state into the desired ground state of H. $\tilde{U}$ is obtained as a quantum circuit with finite depth that depends on a set of parameters $\vec{\phi}$. As discussed in Chapter 1, for any choice of $\vec{\phi}$, the quantum circuit $\tilde{U}(\vec{\phi})$ acting on the product state $|0\rangle^{\otimes n}$ generates a trial wave-function $|\tilde{\varphi}(\vec{\phi})\rangle = \tilde{U}(\vec{\phi})\,|0\rangle^{\otimes n}$. Using a NISQ computer we can compute the expectation value of the energy on that wave-function $E_{\vec{\phi}} = \langle\tilde{\varphi}(\vec{\phi})|\,H\,|\tilde{\varphi}(\vec{\phi})\rangle$. At this stage, we can use a classical optimization algorithm in order to find the values of the parameters $\vec{\phi}$ that minimize the energy, thus providing an approximation to the ground state. The classical optimization allows us to extract

$$\vec{\phi}_{\text{opt}} = \text{argmin}_{\vec{\phi}}\, \langle\tilde{\varphi}(\vec{\phi})|\,H\,|\tilde{\varphi}(\vec{\phi})\rangle \ . \tag{2.1}$$

In this way we can identify $|\varphi_{\text{opt}}\rangle \equiv \tilde{U}(\vec{\phi}_{\text{opt}})\,|0\rangle^{\otimes n}$ with the best possible approximation to the ground state of H, given the architecture of the quantum circuit we can implement on a NISQ device that approximates U.

In order to make contact with practical implementations of VQE, here we will consider unitaries built from a finite set of gates, namely single-qubit rotations and two-qubit controlled operations. We thus are working in the framework of finding the best approximation to a unitary transformation given an elementary set of quantum gates. Given indeed, *e.g.*, m two-body gates that are supposed to act on arbitrary pairs of two qubits out of the n qubits of the systems, we can generate $n(n-1)^m$ in principle distinct quantum circuits.

In order to overcome this exponential scaling, here we take inspiration from perturbation theory. It is known that perturbation theory can be recast in terms of continuous unitary transformations [GW93; Weg94; Gla94; DU04]. In the context of topological order, these continuous unitary transformations have been used to define the quasi-adiabatic continuation [HW05]. Two states are in the same phase if a sequence of gapped local Hamiltonians allows one state to evolve into the other in a finite time. The evolution operator generated by these Hamiltonians, at least in a Trotter approximation, can be represented by a finite-depth quantum circuit [HC15]. These ideas have also been put on firm ground in Ref. [Cir+17], where general theorems about the properties of such unitaries, including their causal cones, have been obtained.

[GW93] Głazek and Wilson, "Renormalization of Hamiltonians"

[Weg94] Wegner, "Flow-equations for Hamiltonians"

[Gla94] Glazek, "Perturbative renormalization group for Hamiltonians"

[DU04] Dusuel and Uhrig, "The Quartic Oscillator: a Non-Perturbative Study by Continuous Unitary Transformations"

[HW05] Hastings and Wen, "Quasiadiabatic continuation of quantum states: The stability of topological ground-state degeneracy and emergent gauge invariance"

[HC15] Huang and Chen, "Quantum circuit complexity of one-dimensional topological phases"

For these reasons, we focus on a fixed geometry of the network represented in Figure 2.1. It can either be interpreted as the Floquet evolution of a local Hamiltonian [KLP18] or as a Trotter approximation of continuous evolution by a local Hamiltonian defined on a line. In 1D, we can separate the terms of the Hamiltonian that act on even and odd links and obtain two sets, each made of mutually commuting gates. A full evolution step involves acting with both sets, and we identify the step with a layer of the circuit. The full ansatz involves concatenating several layers of these unitary gates. As for the particular set of unitaries we consider, they are made out of single-qubit rotations $R_y(\theta)$, and control-Z gates (CZ) that act on two contiguous qubits as shown in Figure 2.1.

A legitimate question is thus how accurate this geometry can be and how close can the state we extract by running a VQE on our set of quantum circuits get to the exact ground state of the system. Since we are dealing with finite systems, the Hamiltonians we are considering always have a gap (at least proportional to $1/n$). If E is the expectation value of the energy on our trial wave-function, its distance from the ground state can be bounded as $\delta \leq \frac{\epsilon}{\Delta E}$ with $\epsilon = E - E_0$ and ΔE being the gap of the Hamiltonian. We will use the error in the ground state energy ϵ to measure the quality of our circuit.

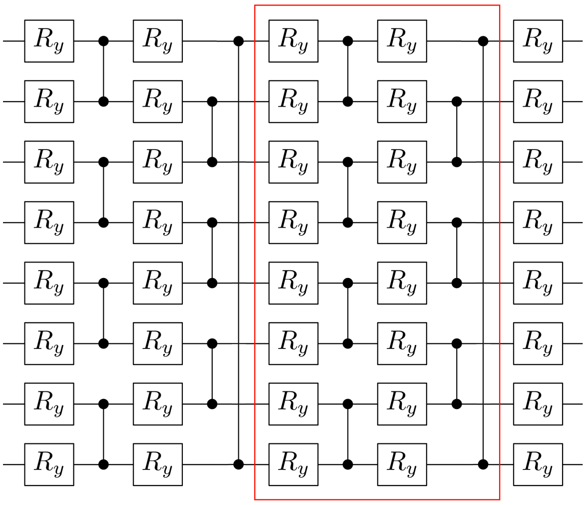

FIGURE 2.1: Variational quantum ansatz for $\tilde{U}(\vec{\phi})$ employed in our simulations. As indicated by the red box, each layer is composed of CZ gates acting on alternating pairs of neighboring qubits which are preceded by $R_y(\theta_i)$ qubit rotations, $R_y(\theta_i) = e^{-i\theta_i Y/2}$. After implementing the layered ansatz, a final layer of $R_y(\theta_i)$ qubit gates is applied. Here, it is shown the case of two layers and $n = 8$ qubits.

2.2 NUMERICAL CHARACTERIZATION

In order to characterize the computational power of the quantum circuit presented in Figure 2.1, as the encoder of the unitary that rotates the initial product state $|0\rangle^{\otimes n}$ into the ground state of a given local Hamiltonian, we need to discuss its entangling power.

In the context of many-body quantum systems, we typically characterize the goodness of a given variational ansatz in terms of how much entanglement it can support. The maximal amount of entanglement that can be generated by our variational quantum circuit acting on a product state depends on its depth. The quantum circuit is indeed built from native unitary gates. Besides single-qubit rotations that do not entangle different partitions, we have one CZ gate per pair of spins at every layer. The CZ is able to create a maximally entangled state between the pair it acts on. For example when acting on $|{+}{+}\rangle$ it transforms it to $\frac{1}{\sqrt{2}}(|0+\rangle + |1-\rangle)$. As a result, and as expected, our unitary quantum circuit can create one bit of entanglement per pair and layer. This fact agrees with the known fact that unitary circuits are able to generate entanglement linearly in their depth as a consequence of Lieb-Robinson bounds [LR72]. A circuit with the structure of the one in Figure 2.1 acting on n constituent made by l layers, indeed could generate up to $min(n/2, l)$ entanglement between two complementary halves of the system made of spins [Cir+17].

[LR72] Lieb and Robinson, "The finite group velocity of quantum spin systems"

[Cir+17] Cirac et al., "Matrix product unitaries: structure, symmetries, and topological invariants"

In the context of ground-state physics, this is a considerable entangling power. In 1D, indeed, ground states of local gapped Hamiltonians full-fill the area-law of entanglement [Has07; ECP10; Laf16], meaning that the entanglement of a block of spins does not grow with the size of the block but rather with the size of its boundaries. In the 1D case we are considering here, no matter how large the bipartition is, if it involves consecutive spins, the boundary is made by just the two sites at each end of the block. As a result, the entanglement of a region of n spins in the ground state of a 1D gapped system asymptotically saturates to a value independent of n. We thus expect that a finite number of layers should be enough to encode the ground state of gapped Hamiltonians of arbitrary number of constituents.

[Has07] Hastings, "An area law for one-dimensional quantum systems"

[ECP10] Eisert et al., "Colloquium: Area laws for the entanglement entropy"

[Laf16] Laflorencie, "Quantum entanglement in condensed matter systems"

When the Hamiltonian is gapless, much less is known since there are theorems stating that the complexity of finding ground states of local translational invariant quantum Hamiltonians is QMA-complete [Aha+09]. However, a special case is the one of gapless Hamiltonians whose ground state can be described by Conformal Field Theories (CFT). In that specific case, we know that the entanglement of a region of $n/2$ contiguous spins in an infinite chain scales asymptotically as

[Aha+09] Aharonov et al., "The power of quantum systems on a line"

$$S(n/2) = \frac{c}{3}\log(n/2) + d\,, \tag{2.2}$$

where c is the central charge of the corresponding CFT and d is a non-universal constant [HLW94; CW94; LRV04; CC04]. In the case of conformally invariant gapless Hamiltonian, we thus expect that the number of layers of our ansatz in Figure 2.1 needs to increase with the size of the system to have a uniform approximation of the system as we increase n, that is in order to accommodate the logarithmic growth of the entropy.

[HLW94] Holzhey et al., "Geometric and renormalized entropy in conformal field theory"

[CW94] Callan and Wilczek, "On geometric entropy"

[LRV04] Latorre et al., "Ground state entanglement in quantum spin chains"

[CC04] Calabrese and Cardy, "Entanglement entropy and quantum field theory"

2.2.1 *The models*

In order to test these expectations, we benchmark the VQE in the case of two paradigmatic quantum spin chains, the Ising model in transverse field and the XXZ chain. Both spin chains are exactly solvable, and we can thus characterize the error in the ground state energy we extract E knowing the exact result for the ground state energy E_0 as $|E - E_0| = \epsilon$. Prior work benchmarking the Quantum Approximate Optimization Algorithm [FGG14] in the Ising model case can also be found in Ref. [MFS19].

[FGG14] Farhi et al., "A quantum approximate optimization algorithm"

[MFS19] Mbeng et al., "Quantum annealing: A journey through digitalization, control, and hybrid quantum variational schemes"

The 1D Ising model is described by the following Hamiltonian

$$H_{Ising} = -\sum_j \sigma^z_j \sigma^z_{j+1} + \lambda \sum_j \sigma^x_j\,, \tag{2.3}$$

where λ is the disordering field. For small λ, the system is in a ferromagnetic phase, where all the spins are aligned along the z direction. As λ increases, the system tends to disorder and goes to a paramagnetic phase for large λ. The two phases are separated by a quantum critical point, exactly at $\lambda = 1$. The system has indeed a $\mathbb{Z}_2$ symmetry generated by $\prod_j \sigma_j^x$ that flips all the spins. The $\mathbb{Z}_2$ symmetry breaks spontaneously at the quantum critical point. In both phases, the elementary excitations are gapped and are spin flips in the paramagnetic phase and domain walls in the ferromagnetic phase. At the critical point, the correct variables to describe the systems are the product of spin and domain walls, giving rise to free Majorana Fermions [SML64]. At the critical point, the system becomes gapless, and the low energy dispersion relation is linear, inducing an emerging Lorentz invariance. The large distance behavior of the transverse field Ising model is described by a CFT with central charge $c = 1/2$, one of the well-known minimal models [BPZ84; Hen99].

[SML64] Schultz et al., "Two-Dimensional Ising Model as a Soluble Problem of Many Fermions"

[BPZ84] Belavin et al., "Infinite conformal symmetry of critical fluctuations in two dimensions"

[Hen99] Henkel, *Conformal Invariance and Critical Phenomena*

From the point of view of entanglement, the ground states of the Ising model in both ferromagnetic and paramagnetic phases are shortly correlated and full-fill the area law. We thus expect that they can easily be generated by a finite-depth quantum circuit, such as the one we are using here.

At the quantum critical point, on the other hand, the ground state violates the area law displaying logarithmic scaling of the entanglement entropy. We thus expect that the number of layers of the circuit needed to keep the accuracy constant increases as we consider increasingly large systems.

The XXZ model is slightly more complicated, and the Hamiltonian reads

$$H_{XXZ} = \sum_j \left(\sigma_j^x \sigma_{j+1}^x + \sigma_j^y \sigma_{j+1}^y \right) + \Delta \sigma_j^z \sigma_{j+1}^z \, , \tag{2.4}$$

where Δ is the spin anisotropy. From the point of view of a Fermionic model, Δ induces a density-density interaction, and thus the model, even if still exactly solvable via the Bethe ansatz, is not anymore a model for free fermions [Ess+05]. For $\Delta \gg 1$ and $\Delta \ll -1$ the system is gapped, and the spins eventually align either ferromagnetically for $\Delta < -1$ and anti-ferromagnetically for $\Delta > 1$ along the z-direction, indicating a Mott-insulating phase for the fermions with either unity filling or checkerboard filling. For values of $-1^+ \leq \Delta \leq 1$ the system is critical, and it describes the physics of a compactified boson, where the radius of compactification depends on Δ. This region is described by a CFT with $c = 1$, and differently from the Ising theory, this one is interacting. Once more, the entropy of a region of consecutive spins in the ground state increases logarithmically with the number of spins in that region, meaning that we expect that the depth of our circuit will have to increase with the system size in order to obtain a uniform accuracy.

[Ess+05] Essler et al., *The One-Dimensional Hubbard Model*

2.2.2 *Gapped Hamiltonians, the perturbative regime*

We now start considering the performance of the VQE in the case of a gapped Hamiltonian. For every realization of the quantum circuit, we run our VQE that selects the optimal values for the free parameters in the circuit that encode the single-qubit rotations around the Y-axis. We have $2n$ parameters per layer that are optimized using a gradient descent method. In particular, the classical method employed in the optimization loop was L-BFGS-B [Byr+95]. This classical method is gradient-based and involves the estimation of the inverse Hessian matrix. We utilized the implemented version of the open-source Python package `SciPy Optimize` [Vir+20], and `QuTiP` [JNN13] for the simulation of the quantum circuits. Parameters are iteratively changed until we reach convergence in the ground state energy. That is, after one iteration, the energy decrease is smaller than a given threshold (typically of the order 10^{-12}).

[Byr+95] Byrd et al., "A Limited Memory Algorithm for Bound Constrained Optimization"

[Vir+20] Virtanen et al., "SciPy 1.0: fundamental algorithms for scientific computing in Python"

[JNN13] Johansson et al., "QuTiP 2: A Python framework for the dynamics of open quantum systems"

Furthermore, we employed standard optimization techniques from tensor networks. In particular, we optimized single-parameters and single-layers, fixing the rest of the trainable elements of the ansatz. We repeated these single-parameter and single-layer optimization cycles until we reached convergence. In addition, we used a recently proposed technique for variational quantum algorithms, called Adiabatically Assisted Variational Quantum Eigensolver (AAVQE) [GSL18].

[GSL18] Garcia-Saez and Latorre, "Addressing hard classical problems with adiabatically assisted variational quantum eigensolvers"

We begin by considering the results for $\lambda = 10$ in Eq. 2.3. We can obtain the ground state in this regime in perturbation theory. We start with the ground state of Eq. 2.3 when $\lambda = \infty$, as the unperturbed state. The Hamiltonian simplifies to $H_0 = \sum_i \sigma_i^x$, whose ground state is a product state in the x basis. We then reduce λ to a finite value, and we can express the ground state of Eq. 2.3 for finite λ perturbatively. The full Hamiltonian can be written as $H = H_0 + \frac{1}{\lambda}\sum_i \sigma_i^x \sigma_{i+1}^x$. Using the perturbation theory in the form of a continuous unitary transformation [DU04], we immediately realize that the l order in the perturbative expansion requires l layers of the quantum circuit. We thus expect that the precision of our ansatz will scale exponentially with the number of layers. Our expectations are confirmed in the numerical results presented in Figure 2.2 where we see very mild size dependence but a clear exponential increase of the accuracy of the VQE with the number of layers for $\lambda = 10$ and $n = 8, 10, 12$.

[DU04] Dusuel and Uhrig, "The Quartic Oscillator: a Non-Perturbative Study by Continuous Unitary Transformations"

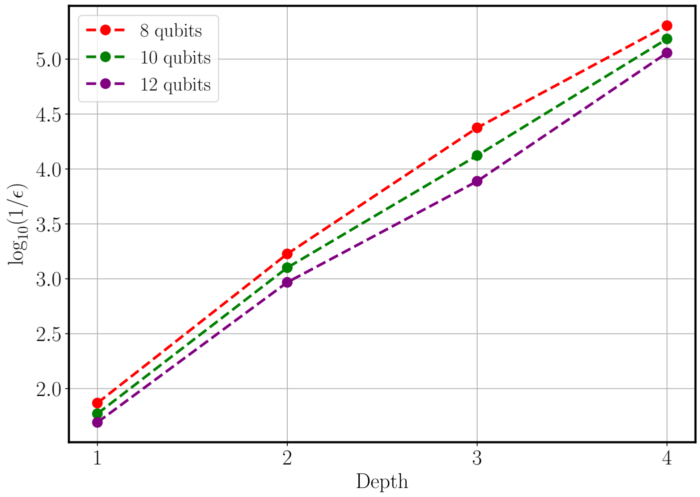

FIGURE 2.2: The error of the ground state energy in logarithmic scale as a function of the number of layers (depth) in the quantum circuit, for the optimal encoding of the ground state of the Ising model in Eq. 2.3 with $\lambda = 10$ for different system sizes $n = 8, 10, 12$. The results lie on straight lines, unveiling an exponential increase of the precision with the number of layers, as expected from a perturbative calculation. For example, with 5 layers, we expect our network could include effects up to $\lambda^{-5} \simeq 10^{-5}$.

In Figure 2.3 we repeat the same analysis as we decrease λ towards the phase transition. For $\lambda = 2$, we still appreciate an exponential scaling of the accuracy, but as expected, the slope of the semi-logarithmic plot is lower since it increases from $1/10$ to $1/2$. For $\lambda = 1$, we appreciate that the behavior of the VQE ground state energy accuracy as a function of the number of layers changes drastically from the behavior observed at larger λ. In the thermodynamic limit, $\lambda = 1$ is the location of the phase transition between the two gapped phases.

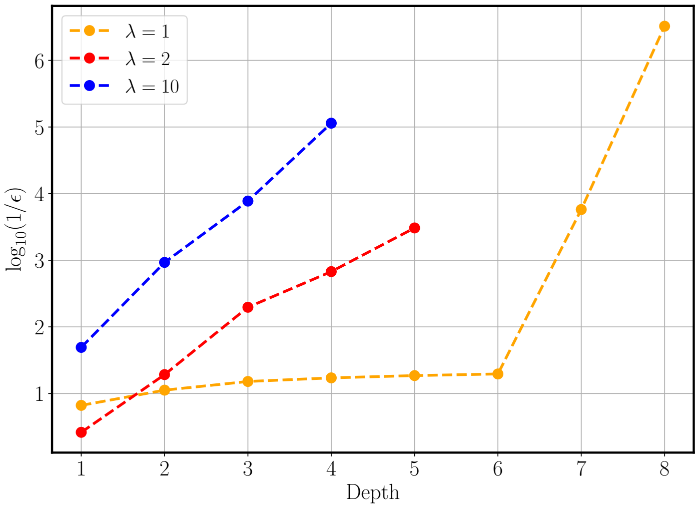

FIGURE 2.3: We benchmark our VQE on a chain of $n = 12$ spins for values of $\lambda = 2, 10$ that are deep in the perturbative regime where the accuracy increases exponentially with the number of layers. At $\lambda = 1$, the Hamiltonian is gapless in the thermodynamic limit. There the accuracy behaves differently as a function of the number of layers, unveiling two regimes.

2.2.3 *Scaling of the accuracy at criticality, the two regimes*

At $\lambda = 1$, the Hamiltonian of a finite length of size n has a gap that closes as $1/n$ and thus becomes gapless in the thermodynamic limit. From the previous discussion about the entangling power of our ansatz, we thus expect that in order to obtain the same accuracy for increasingly large systems, we will need to consider increasingly deep quantum circuits. The amount of entanglement can grow linearly with the depth of the circuit [Cir+17], and in the critical ground state, we only need a logarithmic increase of the entropy. We could thus expect that a number of layers growing logarithmically with the system size could provide a uniform approximation to the ground state of increasingly large systems.

[Cir+17] Cirac et al., "Matrix product unitaries: structure, symmetries, and topological invariants"

In order to verify this expectation, we perform numerical simulations of quantum circuits of several layers (from $l = 1$ to $l = 11$) that are optimized to encode the ground state of systems with different sizes from $n = 6$ to $n = 16$ with periodic boundary conditions. We use the two Hamiltonians in Eqs. 2.3 and 2.4. The Hamiltonians are tuned to a critical point in both cases, choosing $\lambda = 1$ for the Ising model and $\Delta = 1/2$ for the XXZ model. For $\Delta = 1/2$, we are far enough at the same time from the Heisenberg point (where marginally relevant operators tend to make finite-size scaling harder) and from the gapped phases.

In Figure 2.4 we plot the logarithm of the inverse error $\epsilon = |E - E_0|$ versus the depth of the circuit for the Ising model (left) and XXZ model (right). In these plots, the best approximations are points on the far top side of the plot. The accuracy clearly shows two different regimes. Initially, the accuracy varies very little as we increase the number of layers, and hence the number of variational parameters and the entangling power of the circuit. The error indeed stays of the order 10^{-2} from one to several layers for the Ising model and of the order of 10^{-1} for the XXZ model. This behavior seems to be completely independent of the system size since curves obtained by optimizing the energy almost coincide.

In the inset of the two panels of Figure 2.4 we zoom-in in this first regime and plot the same results on a linear scale, that is, we plot $1/\epsilon$ versus l. We can now appreciate that the improvement in accuracy in this regime is a power law of the depth of the circuit, rather than exponential.

We thus seem to observe a *finite-depth* regime, where the precision of the variational scheme depends very little on the number of layers and improves very slowly. This regime changes drastically at a critical number of layers $l = l^*(n)$ that strongly depends on the size of the system. At that critical number of layers, the precision improves several orders of magnitude abruptly. This improvement is particularly abrupt in the case of the Ising model, whereby just adding one layer, the accuracy can improve several orders of magnitude. For the XXZ model, we see similar features though the overall accuracy is lower as a consequence of the higher amount of entanglement in the ground state.

It is interesting to notice that in the *finite-depth* regime, the accuracy in the energy does not depend on the size of the system, differently from what we would expect for finite-size systems, where the energy should approach the thermodynamic limit from below with a correction proportional to $\epsilon \propto 1/n^2$ [Aff86; Car86] for systems with periodic boundary conditions as are the ones we consider here. As the number of layers becomes larger than a critical value of $l^*(n)$ (which once more strongly depends on the system size n), the precision starts to improve exponentially fast with the number of layers. This is consistent with the appearance of a finite correlation length of order n, which is ultimately responsible for the exponential scaling of the energy.

[Aff86] Affleck, "Universal term in the free energy at a critical point and the conformal anomaly"

[Car86] Cardy, "Operator content of two-dimensional conformally invariant theories"

In the XXZ model, the improvement of the energy accuracy when transitioning from the *finite-depth* to the *finite-size* regime is not as sharp as for the Ising model. However, the two regimes are still clearly visible. The first *finite-depth* regime, where the improvement is slow and does not depend on the size of the system but instead on the number of layers, and a *finite-size* regime where the improvement is exponential, and where the slope is different for different system size, revealing the presence of a correlation length that is proportional to the system size. The *finite-size* dominated regime can also be interpreted as a *refinement regime*, since there, with the help of a few additional layers, we typically obtain improvements on the ground state energy of several orders of magnitude.

These two regimes seem to be reminiscent of the finite-entanglement and finite-size regime observed in Matrix Product State (MPS) simulations of the critical systems [Tag+08; Pol+09; Pir+12; Sto+15]. Thus, in order to get a better quantitative characterization of the two regimes, we go back

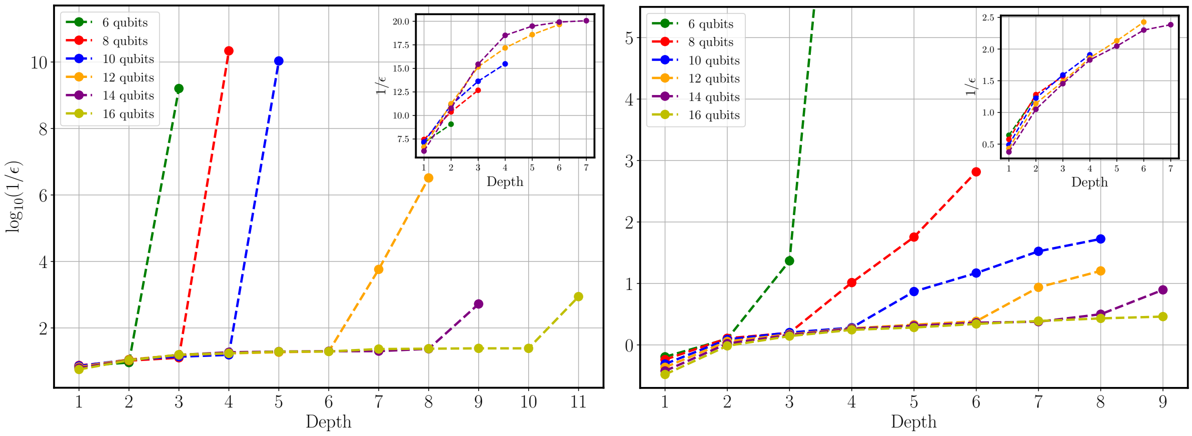

FIGURE 2.4: Error of the ground state energy in logarithmic scale *vs.* number of layers in the variational ansatz, for the Ising model (left) and XXZ model (right). Different colors encode systems made by a different number of qubits n. Better results are encoded by points on the top of the plot, where $\log_{10}(1/\epsilon)$ is large, and hence ϵ is small. As we increase the depth of the circuit, the error initially improves very slowly, as shown by the almost horizontal behavior of the curves. It then suddenly starts to increase several orders of magnitude. This very sharp change of behavior identifies two regimes, namely, *finite-depth* regime, where the energy accuracy does not depend on the size of the system but only on the number of layers and increases slower than exponentially with it, and the *finite-size* regime where the energy accuracy increases exponentially. The insets show a power-law increase of the accuracy in the *finite-depth* regime.

to studying the entanglement entropy of half of the system of the wavefunctions obtained as a result of the VQE.

2.2.4 *Scaling of the entanglement entropy at criticality*

In order to compute the entanglement entropy of the states we obtain from our VQE, we partition the system in two halves each made by $n/2$ contiguous spins. Calling A one of the two halves, we construct the reduced density matrix of A as $\rho_A = \mathrm{tr}_B \left|\varphi_{\mathrm{opt}}\right\rangle \left\langle\varphi_{\mathrm{opt}}\right|$. The Von Neumann entropy of the eigenvalues of ρ_A encodes the entanglement entropy, $S_A = -tr\rho_A \log(\rho_A)$.

Our results for S_A are reported in Figure 2.5. On the left panel, we represent the entropy computed for a bipartition in two halves of the ground state of the Ising model at the critical point. This is obtained by fixing $\lambda = 1$ in the Hamiltonian in Eq. 2.3. We compute the half chain entropy for increasingly large systems from $n = 6$ to $n = 14$ qubits. The entropy shows two regimes. In the first regime, the entropy increases as the number of layers increases. The increase is compatible with being logarithmic in the number of layers, being definitely slower than the linear increase with the number of layers that the circuit could support.

At a critical value of l, $l^*(n)$ that coincides with the critical value observed in the scaling of the energy error, the entropy jumps and saturates to a value that depends on the system size. For values of l larger than $l^*(n)$ the entropy is roughly constant. We can thus fit the entropy as a function of $l^*(n)$, and obtain a good agreement with a scaling of the type $S_A = \alpha \log(l^*(n)) + \beta$. The value of α extracted from a numerical fit to the data is $\alpha_{Ising} = 0.18(2)$. This should be compared with the value of pre-factor that rules the scaling of the entanglement entropy with the size of the system at criticality that only depends on the central charge and is $1/6$. The fact that α is very close to $1/6$ suggests that $l^*(n)$ scales as $l^*(n) = \gamma n$ where γ is a constant.

In the *finite-depth* regime, the entropy of a bipartition is ruled by the number of layers of the VQE rather than by the size of the bipartition. We observe a logarithmic increase of the entropy with the number of layers. The entropies of sub-regions with a very different number of constituents are very similar. A fit of the data using a logarithmic increase of the entropy as a function of the number of layers is plotted in black, both for the Ising and for the XXZ model, in Figure 2.5. Even if the corresponding curve significantly deviates from the numerical value for large depths, it correctly reproduces the average values of the entropies for different systems sizes obtained with VQE having the same number of layers l in the regime where $l \ll l^*(n)$. It is important to notice that obtaining accurate values for the entropy in the *finite-depth* regime is very challenging. We are indeed optimizing the energy in a manifold of excited states, where the energy is still considerably higher than the energy of the ground state. As a result, there are many states with roughly the same energies but very different entropies, as observed in the context of MPS simulations [Tag+08]. The fit to the entropy as a function of the logarithm of the depth of the circuit, in the *finite-depth* regime, that is for $l \ll l^*(n)$ provides a value for $\alpha = 0.13(4)$. The number is compatible with the expected scaling of the entropy of the system that deviates from the CFT due to the presence of a finite correlation length

[Tag+08] Tagliacozzo et al., "Scaling of entanglement support for matrix product states"

$$\xi(l) \propto l\,. \tag{2.5}$$

The right panel of Figure 2.5 presents a similar study of entanglement entropy in the ground state of the XXZ model described by the Hamiltonian in Eq. 2.4. The behavior is similar to the one observed in the case of the Ising model at the critical point. However, we appreciate a much larger entropy as expected from the fact that the central charge of this model is twice the one for the Ising model, *i.e.*, $c = 1$. Once more, we observe two regimes, one regime where the entropy is roughly independent on the size of the partitions but depends strongly on l and seems to follow a logarithmic increase. At the values of $l = l^*(n)$ already identified in Figure 2.4, we see that the entropies jump to values that depend on the size of the block. For larger values of l, the entropy remains almost constant. We identify $l^*(n)$ with the last point of each numerical series presented in Figure 2.5. We can fit a logarithmic growth of the entropy at that specific value of $l^*(n)$, obtaining for the coefficient of the logarithmic scaling $\alpha_{XXZ} = 0.37(9)$. This value is compatible with $c/3 = 1/3$. Comparing this result with the expected scaling for the entropy of a bipartition made by $n/2$ spin in Eq. 2.2, we have a further indication that $l^*(n) = \gamma n$ with γ constant. Further confirmation of this identification can be obtained by fitting the lower part of the numerical sequences for the entropies as a function of $\log(l)$. In these regions, for $l \ll l^*(n)$, the entanglement entropy depends only mildly on $n/2$, the size of the partitions. The result of the best fit tells us once more that $S_A = \alpha \log(l) + \beta$ with $\alpha_{XXZ} = 0.24(16)$. Once more, this result is compatible with the $1/3$ expected for a system having effective length l rather than $n/2$ thus providing further confirmation of Eq. 2.5.

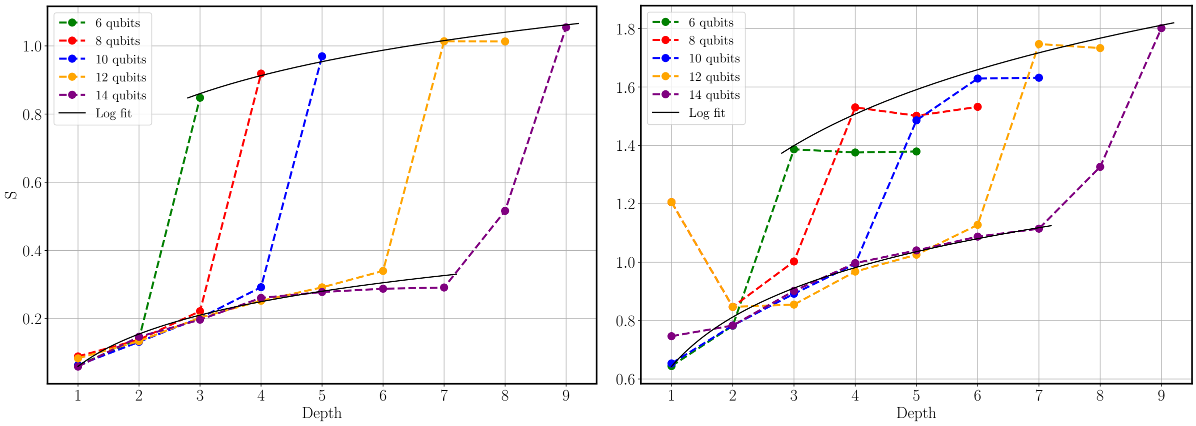

FIGURE 2.5: Von Neumann entropy of the bipartition *vs.* the number of layers in the variational ansatz, for the Ising model (left) and XXZ model (right), and for increasing number of qubit n. Black lines represent logarithmic fits of the data. Once more, the sudden growth of the entropy coincides with the change of regime.

2.3 OUTLOOK

In the previous sections, we have unveiled that a VQE that uses the structure of the quantum circuit presented in Figure 2.1 is able to accurately describe the ground state of local Hamiltonians both in gapped regimes and in those gapless regimes that can be described by a CFT. This, by itself, is an important observation given the current availability of NISQ devices in the labs.

As we have discussed, the circuit structure in Figure 2.1 is inspired by the idea of quasi-adiabatic continuation [HW05], a set of analytical results that tell us that whenever two states belong to the same phase, we can transform one into the other by evolving it using a local gapped Hamiltonian for a finite amount of time. The resulting finite time evolution, at least in a Trotter approximation, takes the form of the tensor network in Figure 2.1. Analogously when two states are in the same phase, we can build one from the other by applying to it a perturbation. The corresponding perturbative expansion can also be casted as a continuous unitary transformation [GW93; Weg94; Gla94; DU04] that can be discretized and expressed as the circuit in Figure 2.1 [Van+17]. From this perspective, the results we have presented concerning the performances of the VQE in the gapped regime are not surprising. However, they confirm that whenever we can rotate the ground state of a gapped Hamiltonian into a product state, our circuit, made of simple elementary gates, does it optimally, with a precision that improves exponentially with its depth, as expected from perturbation theory.

The results in the critical regime are much more interesting. First of all, by definition, the critical point is not connected to a product state via a perturbative expansion. However, there is no true critical point in a finite-size system; thus, the fact that we can encode faithfully pseudo-critical finite systems has to be expected. Our results go beyond this expectation and allow us to identify the minimum depth of the circuit required to represent the pseudo-critical grounds state of a finite system faithfully.

Our numerical results for critical systems point to the existence of two different regimes. A regime that we have called *finite-depth* in which $l < n/2$

[HW05] Hastings and Wen, “Quasiadiabatic continuation of quantum states: The stability of topological ground-state degeneracy and emergent gauge invariance”

[GW93] Głazek and Wilson, “Renormalization of Hamiltonians”

[Weg94] Wegner, “Flow-equations for Hamiltonians”

[Gla94] Glazek, “Perturbative renormalization group for Hamiltonians”

[DU04] Dusuel and Uhrig, “The Quartic Oscillator: a Non-Perturbative Study by Continuous Unitary Transformations”

[Van+17] Vanderstraeten et al., “Bridging Perturbative Expansions with Tensor Networks” where the precision of the results only depends on the number of layers and a *refinement*, or *finite-size* regime. In the *finite-depth* regime, the accuracy of the ground state energy increases very slowly with the number of layers, only polynomially. The entanglement entropy of a region that, in CFT, should increase logarithmically with the number of spins in that region only increases logarithmically with the number of layers of the circuit. Two half-system bipartitions taken from systems with different sizes (and hence containing a different number of spins) have roughly the same entropy when they are computed from a quantum circuit with depth smaller than $l \ll l^*(n)$.

In the *refinement* regime, the results acquire the expected finite-size dependence. In that regime, the precision increases exponentially with the number of layers, as seen from straight lines in Figure 2.4. The slope of the straight lines allows us to define a correlation length ξ as $\epsilon \propto \exp(-l/\xi)$ that as clear from the plots depends on n, as $\xi \propto n$. As expected in the *finite-size* regime, the entanglement of a region made by $n/2$ spins increases logarithmically with $n/2$.

The logarithmic increase of the entropy in the *finite-depth* regime with a pre-factor that is numerically compatible with the ones dictated by the CFT and the location of the jump between the two regimes at a value $l^*(n) \propto n$ are compatible with the following analysis. The finite depth of the system induces an effective correlation length $\xi_l \propto l$ as described by our main result in Eq. 2.5. Since the finite size of the system also induces a finite correlation length $\xi_n \propto n$ we can expect a cross-over phenomenon when $l \simeq n$, where the system transition from a regime in which the shortest length is the one induced by l to a regime where finite-size effects become dominant since the shortest correlation length is the one induced by the size of the system.

Possibly the most interesting observation is the fact that in the *finite-depth* regime, the correlation length is proportional to l as encoded by Eq. 2.5. This is a direct consequence of Lieb-Robinson bounds [LR72; BHV06]. Indeed if we think of l as the computation time, we immediately understand that, as a consequence of the existence of a finite speed of propagation, in order to build up correlations at a distance l, we need to wait for times proportional to l. This simple explanation is compatible with what we observe numerically. This behavior was further verified in Refs. [Con+21; JSP22].

[LR72] Lieb and Robinson, “The finite group velocity of quantum spin systems”

[BHV06] Bravyi et al., “Lieb-Robinson Bounds and the Generation of Correlations and Topological Quantum Order”

3

Data encoding in quantum autoencoders

> *"With four parameters I can fit an elephant, and with five I can make him wiggle his trunk."*
> —John von Neumann

A limiting factor for most of the near-term quantum computing applications is the amount of quantum resources that can be realized in current experiments. Indeed, for noisy intermediate-scale quantum (NISQ) [Pre18] devices, any tool that can reduce the amount of quantum resources can be considered especially valuable. For classical data processing, autoencoders are standard tools used to reduce the resources [Kra91].

[Pre18] Preskill, "Quantum Computing in the NISQ era and beyond"

[Kra91] Kramer, "Nonlinear principal component analysis using autoassociative neural networks"

A classical autoencoder is an artificial neural network used to learn a representation of a data set in a lower-dimensional space, typically known as latent dimension. This encoding is validated by attempting to reproduce the input data from the lower-dimensional data. If the autoencoder successfully achieves this task, at least approximately, the network has learned to encode the original information into the smaller, latent space.

In the following section, we introduce the concept of a quantum autoencoder (QAE) [ROA17], which is inspired by the classical autoencoder design. The motivation for a QAE is to be able to recognize patterns beyond the capabilities of a classical autoencoder, given the different properties of quantum mechanics, namely, superposition and entanglement. This model will allow us to perform analogous machine learning tasks for quantum systems and reduce the required quantum resources when dealing with quantum data. For instance, quantum autoencoders could be used as a state preparation engine in the context of variational quantum algorithms. That is, we could combine, say, a Variational Quantum Eigensolver (VQE) [Per+14] with a quantum autoencoder, where now the only active VQE parameters are associated with the latent space.

[ROA17] Romero et al., "Quantum autoencoders for efficient compression of quantum data"

[Per+14] Peruzzo et al., "A variational eigenvalue solver on a photonic quantum processor"

3.1 QUANTUM AUTOENCODERS

The quantum autoencoder model is analogous to that of a classical autoencoder. The QAE network is designed to compress the input quantum information through a smaller latent space. In this scheme, we look for a parameterized quantum circuit $U(\vec{\theta})$ that encodes an initial input state into an intermediate latent space, after which the action of the decoder, $U^{\dagger}(\vec{\theta})$, attempts to reconstruct the input. Indeed, for a quantum circuit to resemble an autoencoder network, the information contained in some of the qubits

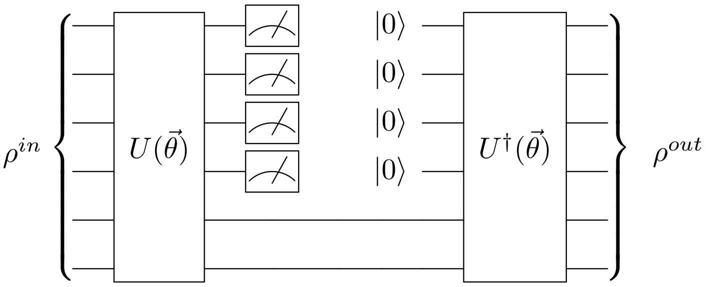

FIGURE 3.1: Circuit implementation of a 6-qubit quantum autoencoder with a 2-qubit latent space, that is, $n = 2$ and $k = 4$. The unitary $U(\vec{\theta})$ encodes a 6-qubit input state ρ^{in} into a 2-qubit intermediate state, after which the decoder $U^\dagger(\vec{\theta})$ attempts to reconstruct the input, resulting in the output state ρ^{out}.

must be discarded after the initial encoding $U(\vec{\theta})$. New qubits, which are initialized to some reference state, are then used to implement the decoding action $U^\dagger(\vec{\theta})$. A graphical depiction can be seen in Figure 3.1.

Let us be more specific with the quantum autoencoder formulation. Let ρ^{in} be an ensemble of pure states on $n+k$ qubits, where k qubits are going to be discarded after the action of the encoder. Let $U(\vec{\theta})$ be a unitary operation acting on the $n+k$ qubits, where $\vec{\theta}$ is a set of classical parameters defining the quantum circuit. Also let $|0\rangle^{\otimes k}$ the pure reference state of k qubits. Using machine-learning techniques, we want to find the best unitary $U(\vec{\theta})$ which, on average, best preserves the input state ρ^{in}. The cost function to minimize is

$$C(\vec{\theta}) = 1 - F(\rho^{in}, \rho^{out}), \tag{3.1}$$

where $F(\rho^{in}, \rho^{out})$ is the fidelity of the input and output state after the action of the decoder. Notice that if $C(\vec{\theta}) = 0$, then we accomplished the learning task of finding the ideal $U(\vec{\theta})$, and therefore, we achieved a lossless compression, *i.e.*, $\rho^{in} = \rho^{out}$. For readers interested in experimental applications, a QAE implementation in a photonic device can be seen in Ref. [PTP19].

Recently, much attention has been paid to data encoding in variational quantum algorithms [Llo+20; LC20] since it was proven that data encoded into the model alters the expressive power of parameterized quantum circuits [SSM21; GTN21]. Specifically, this idea has been implemented for classification of data [Hav+19; PS+20a], and to study energy profiles of quantum Hamiltonians [CLKAG21]. The following sections explore how data encoding influences a quantum autoencoder by introducing the enhanced feature quantum autoencoder. As we will see, its key ingredient is to include a feature vector into the variational quantum circuit that characterizes the model we aim to compress.

[PTP19] Pepper et al., “Experimental realization of a quantum autoencoder: The compression of qutrits via machine learning”

[Llo+20] Lloyd et al., “Quantum embeddings for machine learning”

[LC20] LaRose and Coyle, “Robust data encodings for quantum classifiers”

[SSM21] Schuld et al., “Effect of data encoding on the expressive power of variational quantum-machine-learning models”

[GTN21] Goto et al., “Universal Approximation Property of Quantum Machine Learning Models in Quantum-Enhanced Feature Spaces”

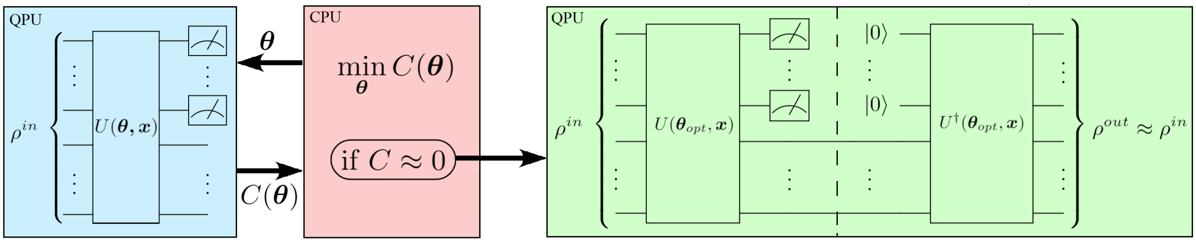

FIGURE 3.2: Schematic representation of the EF-QAE. The input to EF-QAE is a set of initial states ρ^{in}, a feature vector $\vec{x}$ that characterizes the initial states, and a shallow sequence of quantum gates U. The feature vector $\vec{x}$ is encoded together with the variational parameters $\vec{\theta}$, where the latter are adjusted in a quantum-classical optimization loop until the local cost $C(\vec{\theta})$ converges to a value close to 0. When this loop terminates and the optimal parameters $\vec{\theta}_{opt}$ are found, the resulting circuit $U(\vec{\theta}_{opt}, \vec{x})$ prepares compressed states $|\phi\rangle$ of a particular model. Moreover, we may apply $U^{\dagger}(\vec{\theta}_{opt}, \vec{x})|0\ldots0\rangle \otimes |\phi\rangle$ to recover $\rho^{out} \approx \rho^{in}$.

3.2 ENHANCED FEATURE QUANTUM AUTOENCODER

3.2.1 *Overview*

Here, we present the enhanced feature quantum autoencoder (EF-QAE). A schematic diagram of the EF-QAE can be seen in Figure 3.2. The algorithm can be initialized with a set of initial states ρ_i^{in}, a feature vector $\vec{x}$, and a shallow sequence of quantum gates U. In this scheme, we define a unitary $U(\vec{\theta}, \vec{x})$ acting on the initial state ρ_i^{in}, where $\vec{x}$ is a feature vector that characterizes the set of input states. For instance, as we will see in Section 3.3, $\vec{x}$ may be the transverse field λ of the 1D Ising spin chain. Once the trial state is prepared, measurements are performed to evaluate the cost function $C(\vec{\theta})$. This result is then fed into the classical optimizer, where the parameters $\vec{\theta}$ are adjusted. This quantum-classical loop is repeated until the cost function converges to a value close to 0. When the loop terminates, $U(\vec{\theta}_{opt}, \vec{x})$ prepares compressed states $|\phi\rangle$ of a particular model.

A summary comparing EF-QAE and QAE proposed in Ref. [ROA17] can be seen in Table 3.1. Note that the main difference between EF-QAE and QAE is the presence of a feature vector $\vec{x}$ in the sequence of gates U. This will allow us to study and explore how data encoding influences the behavior of a quantum autoencoder.

[ROA17] Romero et al., “Quantum autoencoders for efficient compression of quantum data”

	QAE	**EF-QAE**
Quantum resources (circuit depth)	Equal	
Unitary operation	U($\vec{\theta}$)	U($\vec{\theta}, \vec{x}$)
No. trainable parameters (in each rotation gate)	1	2
Classical optimization	EF-QAE generally needs additional optimization	
Compression performance	EF-QAE has always higher compression performance	

TABLE 3.1: Summary for QAE and EF-QAE similarities and differences.

3.2.2 *Cost function*

The goal of a quantum autoencoder is to store the quantum information of the input state through the smaller latent space. Therefore, it is important to quantify how well the information is preserved. This, in general, is quantified by a cost function that one has to minimize. As we discussed in Section 3.1, in Ref. [ROA17], this cost function evaluates the fidelity of the input and output states, and it is constructed from global operators. However, it is known that global cost functions lead to trainability issues even for shallow depth quantum circuits [McC+18; Cer+21b].

[ROA17] Romero et al., “Quantum autoencoders for efficient compression of quantum data”

[McC+18] McClean et al., “Barren plateaus in quantum neural network training landscapes”

[Cer+21b] Cerezo et al., “Cost function dependent barren plateaus in shallow parametrized quantum circuits”

To address this issue, we use a cost function designed from local operators as proposed in Ref. [Cer+21b]. There is a close connection between data compression and decoupling. That is, if the discarded qubits, from now on referred to as trash qubits, can be perfectly decoupled from the rest, the autoencoder reaches lossless compression. For instance, if the output of the trash subsystem is a fixed pure state, say $|0\rangle^{\otimes k}$, then it is decoupled, and consequently, the input state has been successfully compressed.

A figure of merit to quantify the degree of decoupling, or data compression, when training is simply the total amount of non-zero measurement outcomes on the k trash qubits, which will be minimized. To design the cost function to be local, different outcomes may be penalized by their Hamming distance to the reference $|0\rangle^{\otimes k}$ state, which is just the number of symbols that are different in the binary representation. Thus, the local cost function C to be minimized is

$$C \equiv \sum_{m,j} d_{Hj} M_{m,j} \equiv \frac{1}{2}\sum_{m=1}^{k}(1-\langle Z_m\rangle)\,, \tag{3.2}$$

where d_{Hj} denotes the Hamming distance and $M_{m,j}$ are the results of the j-th measurement on the m trash qubit in the computational basis. Equivalently, it can also be defined in terms of local Z Pauli operators. Notice that this cost function delivers direct information on how the compression of the trash qubits is performed and has a zero value if and only if lossless compression is achieved.

3.2.3 *Ansatz*

To implement the EF-QAE model on a quantum computer, we must define the form of the parametrized unitary $U(\vec{\theta},\vec{x})$, decomposing it into a quantum circuit suitable for optimization. Recall that a quantum autoencoder may be thought of as a disentangling unitary. The complexity of the circuit thus limits this property. Given the limited available quantum resources in practice, due to the coherence times and gate errors, we will look for a circuit structure that maximally exploits entanglement while maintaining a shallow depth.

A primitive strategy to construct a variational circuit in a more general case may consist of building a circuit of arbitrary 2- and 1-qubit gates characterized by some parameters. However, this is a naive approach. The action of the EF-QAE on the original state is

$$U|\psi\rangle = |0\rangle \otimes \ldots \otimes |0\rangle \otimes |\phi\rangle\ . \tag{3.3}$$

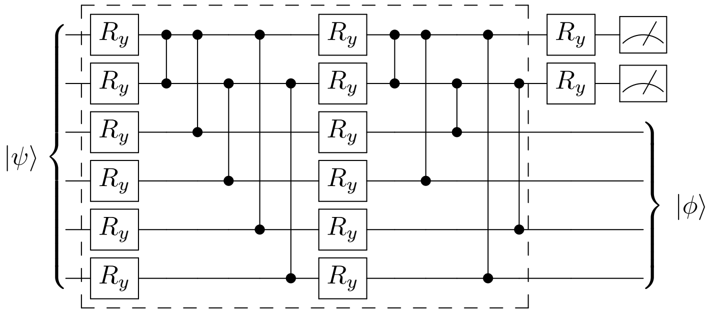

FIGURE 3.3: Variational quantum ansatz employed for the EF-QAE model. As indicated by the dashed box, each layer is composed of CZ gates acting on the trash qubits preceded by R_y qubit rotations, $R_y(\theta_j) = e^{-i\theta_j Y/2}$. A cascade of CZ gates is then applied between the trash qubits and the qubits containing the final compressed state. After implementing the layered ansatz, a final layer of R_y qubit gates is applied to the trash qubits. Note that the sequence of entangling gates can be applied mostly in parallel.

Thus, it is clear that the entangling gates should primarily act between each of the trash qubits, and between the trash qubits and the qubits containing the final compressed state. Subsequently, we may avoid using entangling gates between the qubits that are not trash while maximizing the entangling gates on the ones of interest. This could be done using a similar structure to that depicted in Figure 3.3. Notice that most of the sequence of entangling gates can be applied in parallel at the same step and that the number of quantum gates is linear with the number of qubits and layers.

The feature vector $\vec{x}$ is encoded into each of the single R_y qubit rotations by using a linear function as

$$R_y\left(\vec{\theta},\vec{x}\right) = R_y\left(\theta^{(i)}x^{(j)} + \theta^{(i+1)}\right), \tag{3.4}$$

where i, j indicates a component of the vector, and $\vec{\theta}$ are the parameters adjusted in the optimization loop. The rationale behind choosing this kind of encoding is that it has been shown to provide universality, provided enough layers, and with a single qubit [PS+20a]. Although we use multiple qubits, and entanglement is allowed, we expect a similar behavior as the number of layers increases. Note that this encoding is analogous to that used in classical neural networks. That is, $\vec{\theta}$ plays the role of the weights and biases, while the rotation gate plays the role of the non-linear activation function. On the other hand, the role of the feature vector $\vec{x}$ is inspired by feed-forward classical neuronal networks. Specifically, in this type of classical network, data is reintroduced and processed by many layers of neurons, similar to what our quantum circuit is doing. Other encoding strategies of the feature vector can be considered, for instance, using a non-linear encoding [CLKAG21].

[PS+20a] Pérez-Salinas et al., “Data re-uploading for a universal quantum classifier”

[CLKAG21] Cervera-Lierta et al., “Meta-variational quantum eigensolver: Learning energy profiles of parameterized hamiltonians for quantum simulation”

From a quantum mechanical perspective, let us provide the mathematical intuition on why the EF-QAE may achieve a better compression performance. Consider a set of states, $\rho = \sum_i p_i \left|\psi_i\right\rangle$, with support on a subset of a Hilbert space $\mathcal{S} \subset \mathcal{H}$. Note that the dimension of $\mathcal{S}$ is dictated by the rank of the matrix ρ. The action of a standard QAE is to apply 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 $\log_2|\mathcal{H}-\mathcal{S}|$. It is key to note that unitary operations (the action of the encoder) do not modify the rank of the matrix ρ, and therefore, the compression is initially limited by its rank. In contrast, the action of the

EF-QAE can be seen overall as a non-unitary operation in ρ, since the action of the encoder is tailored to a particular input $|\psi_i\rangle$, informed by the feature vector $\vec{x}$. That is, EF-QAE is applying different unitary operations $U(\vec{\theta},\vec{x})$ to different input states $|\psi_i\rangle$, depending on the extra information delivered by the feature vector $\vec{x}$, and by doing so, modifying the rank of the matrix ρ. This allows the EF-QAE to improve the compression performance.

3.3 NUMERICAL SIMULATION: 1D ISING SPIN CHAIN

The EF-QAE can be verified on simulations. We utilized the open-source Python API `Qibo` [Eft+20; Eft+21b] for the simulation of the quantum circuits. Here, we benchmark both the EF-QAE and the standard QAE in the case of a paradigmatic quantum spin chain with 6 qubits, the transverse field Ising model. The 1D Ising model is described by the following Hamiltonian

[Eft+20] Efthymiou et al., "Quantum-TII/qibo on Github"

[Eft+21b] Efthymiou et al., "Qibo: a framework for quantum simulation with hardware acceleration"

$$H_{Ising} = \sum_j \sigma_j^z \sigma_{j+1}^z + \lambda \sum_j \sigma_j^x\,, \tag{3.5}$$

where λ is the transverse field. In the thermodynamic limit, the system has a quantum phase transition exactly at $\lambda = 1$.

The EF-QAE and QAE are optimized over a training set of ground states of the Ising model. Specifically, we have considered N=20 equispaced ground states in between $\lambda = 0.5$ and $\lambda = 1.0$, with initial random parameters. For the cost function, we computed Eq. 3.2 for each training state and then averaged them as

$$C_N = \frac{\sum_i^N C_i}{N}\,. \tag{3.6}$$

Nonetheless, notice that for other models, sophisticated cost functions could be more convenient to implement. We have considered the variational quantum circuit in Figure 3.3 with 3 layers, and therefore, the resulting compressed state contains 4 qubits. Here, the feature vector $\vec{x}$ for the EF-QAE is a scalar that takes the value of the transverse field λ.

The classical technique employed in the optimization loop is the BFGS method, which is gradient-based and involves estimation of the inverse Hessian matrix [NW06]. Let us also briefly comment here on the training required for both QAE and EF-QAE. Indeed, although the depth of the circuit is equivalent, the number of trainable parameters is not. In this sense, QAE has 1 trainable parameter on each rotation-gate, whereas EF-QAE has 2 trainable parameters. This may imply the computation of extra gradients for gradient-based optimizers and, therefore, extra cost function evaluations. Recall, however, that this possible classical overhead is only present during the training procedure, and hence, we will not face any overhead when using a pretrained EF-QAE in combination with other machine learning tasks.

[NW06] Nocedal and Wright, *Numerical optimization*

In Figure 3.4, we show the cost function value as a function of the number of evaluations. The EF-QAE* is the EF-QAE initialized with the optimal parameters of QAE. This way, the EF-QAE* will always improve the QAE performance. As can be seen, the EF-QAE achieves almost twice the compression of the QAE. Nevertheless, notice that for the EF-QAE, the number of function evaluations required to achieve higher compression is larger. Recall that this is simply a trade-off between classical and quantum

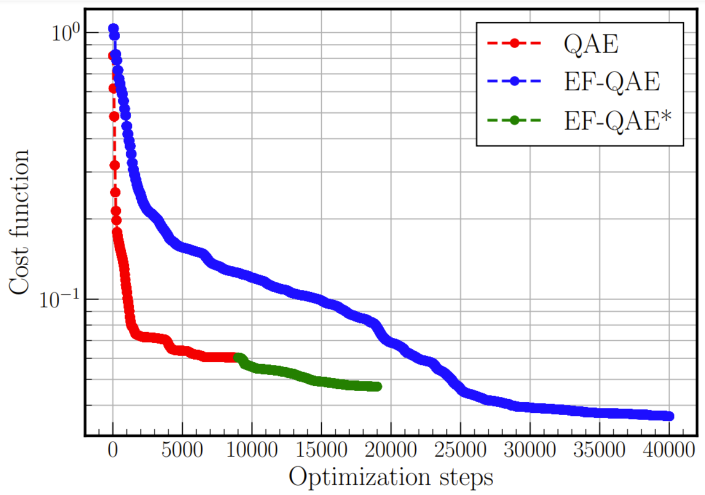

FIGURE 3.4: Cost function value as a function of the number of evaluations. Here, we consider the standard QAE, EF-QAE, and EF-QAE*. The EF-QAE* is the EF-QAE initialized with the optimal parameters of QAE. The EF-QAE achieves almost twice the compression of the QAE using the same quantum resources, at the expense of additional classical optimization.

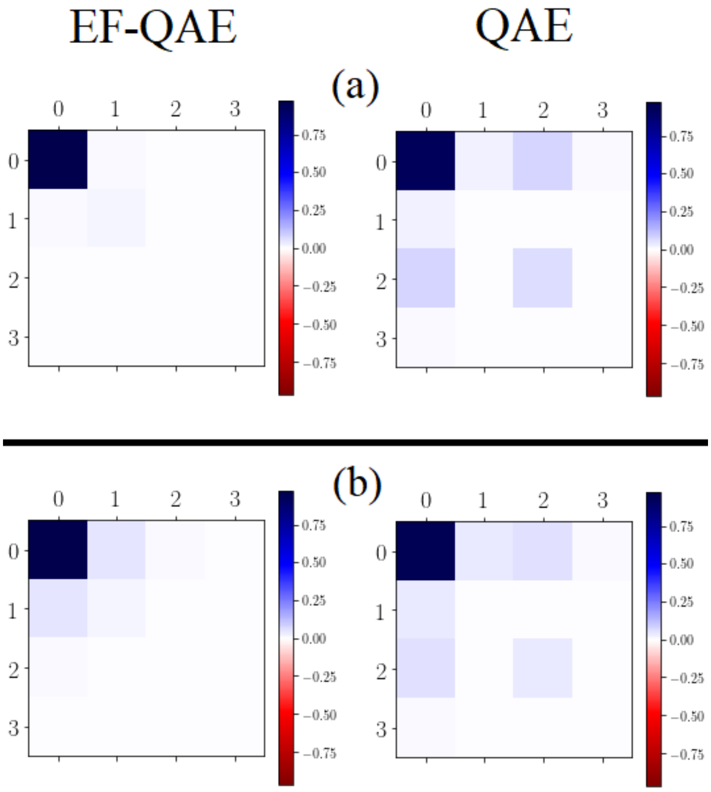

FIGURE 3.5: Visualization of the trash space for the EF-QAE and QAE, considering two different test ground states of the 1D Ising model corresponding to (a) $\lambda = 0.60$ and (b) $\lambda = 0.75$. The size of the register is two qubits. The space is characterized as the density matrix of the trash state. Integer labels denote the binary representation of the computational basis states.

resources. Using the same quantum resources, we improve the compression performance at the expense of additional classical optimization.

To quantify these expectations, we assess both EF-QAE and QAE with the optimal parameters against two test ground states of the Ising model, specifically, with $\lambda = 0.60$ and $\lambda = 0.75$. The results are shown in Figure 3.5. Here, we show a density matrix visualization of the trash space. The EF-QAE achieves better compression to the $|00\rangle$ trash state and, therefore, higher fidelity on the output state. As we change the values of the transverse field, we note that compression differs. In Figure 3.6 we show the output fidelities of 20 training and 60 test Ising ground states. As can be seen, the output fidelities of the EF-QAE are higher, except for a few outlier values around $\lambda = 0.7$. This could be improved, for instance, by simply increasing the number of training states or by populating values around $\lambda = 0.7$ taking nonequispaced training ground states.

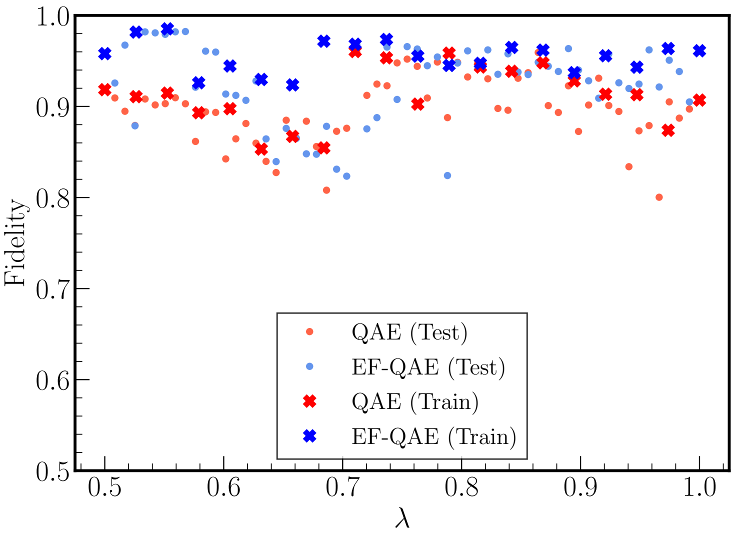

FIGURE 3.6: Fidelity of the output state for Ising ground states with different transverse field λ, using the EF-QAE and QAE. We have considered 20 training and 60 test ground states.

3.4 OUTLOOK

We have presented a variational quantum algorithm called EF-QAE capable of compressing quantum data of a parameterized model. In contrast to standard QAE, EF-QAE achieves this compression with higher fidelity. Its key idea is to define a parameterized quantum circuit that depends upon adjustable parameters and a feature vector that characterizes such a model. In this way, the data compression can be tailored to the particular input, informed by the feature vector, and the compression performance is enhanced.

We have validated the EF-QAE in simulations by compressing ground states of the 1D Ising spin chain. We compared the results with the standard QAE. The results show that EF-QAE achieves better compression of the initial state, and therefore, the final output state is recovered with higher fidelity. Moreover, the learning task of EF-QAE can be initialized with the optimal QAE parameters. In this manner, EF-QAE will always improve the QAE performance. The encoding strategy of the feature vector is amenable to be improved, for instance, using a non-linear encoding. We leave the study of encoding strategies for future works.

The EF-QAE may need additional classical optimization compared to QAE. In contrast, we increase the compression performance using the same amount of limited quantum resources. In this sense, EF-QAE is a step toward what could be done on NISQ computers, shortening the distance between current quantum devices and practical applications.

4

Quantum singular value decomposition

> "*If I feel unhappy, I do mathematics to become happy. If I am happy, I do mathematics to keep happy.*"
> —Alfréd Rényi

Much progress has been made towards a better understanding of bipartite and multipartite entanglement of quantum systems in the last decades. Among the many figures of merit that have been put forward to quantify entanglement, the von Neumann entropy stands out as it finely reveals the quantum correlations between subparts of the system. Yet, the explicit computation of this entropy and many other bipartite measures of entanglement rely on a clever decomposition of the tensor that describes a two-party system. On the experimental side, although entropies remain elusive as no direct observable straightforwardly describes them, a few approaches have been proposed in ultracold atoms [AD12; Isl+15].

The fundamental mathematical tool to analyze bipartite entanglement is the so-called Schmidt decomposition [EK95; Per06], also named as Singular Value Decomposition (SVD). Given the knowledge of a bipartite state in its tensor form, the SVD casts this tensor onto a simpler diagonal form, which unveils the entanglement structure of the original state. In practice, the SVD requires the knowledge of the coefficients of the state and needs further computational effort to get the eigenvalues and eigenvectors that fully characterize the state.

A few proposals have been put forward for diagonalizing a matrix on a quantum computer. One of them relies on exponentiation of the matrix, and subsequent application of the Quantum Phase Estimation procedure [Reb+18]. The second proposal [LaR+18] is a variational algorithm that seeks to directly diagonalize a density matrix ρ by simultaneously acting on two copies of the quantum state described by ρ. In this case, the cost function to be minimized quantifies how far the state is from being diagonal in terms of purity. There also exist quantum algorithms [JST17; SCC19] that compute Rényi entropies, and from them, the largest eigenvalues of reduced density matrices. Finally, a different approach using a continuous-variable quantum computer is considered in [DSW18].

In this chapter, we present a quantum circuit that produces the elements of the SVD of a pure bipartite state that we shall call QSVD for Quantum Singular Value Decomposer. As discussed in the following sections, the proposed circuit is made of two unitaries, each acting on a separate subpart of the system that can be determined in a variational way. The frequencies

[AD12] Abanin and Demler, "Measuring entanglement entropy of a generic many-body system with a quantum switch"

[Isl+15] Islam et al., "Measuring entanglement entropy in a quantum many-body system"

[EK95] Ekert and Knight, "Entangled quantum systems and the Schmidt decomposition"

[Per06] Peres, *Quantum theory: concepts and methods*

[Reb+18] Rebentrost et al., "Quantum singular-value decomposition of nonsparse low-rank matrices"

[LaR+18] LaRose et al., "Variational quantum state diagonalization"

[JST17] Johri et al., "Entanglement spectroscopy on a quantum computer"

[SCC19] Subaşı et al., "Entanglement spectroscopy with a depth-two quantum circuit"

[DSW18] Das et al., "Continuous-variable quantum gaussian process regression and quantum singular value decomposition of nonsparse low-rank matrices" of the outputs of the final state in the circuit deliver the eigenvalues of the decomposition without further treatment. Also, the eigenvectors of the decomposition can be recreated from direct action of the adjoint of the unitaries that conform the system on trivial states.

The key ingredient of the algorithm is to train the circuit on the exact coincidence of outputs. This is a subtle way to force a diagonal form onto the state. It also provides an example of a quantum circuit that is not trained to minimize some energy but rather to achieve a precise relation between the superposition terms in the state (other examples can be found in Refs. [ROA17; Car+20; PS+20b; Szo+22]). We further verify the QSVD algorithm on simulations.

[ROA17] Romero et al., "Quantum autoencoders for efficient compression of quantum data"

[Car+20] Carolan et al., "Variational quantum unsampling on a quantum photonic processor"

[PS+20b] Pérez-Salinas et al., "Measuring the tangle of three-qubit states"

[Szo+22] Szołdra et al., "Unsupervised detection of decoupled subspaces: many-body scars and beyond"

A peculiar bonus of this approach is that the QSVD provides a means to perform a SWAP between parties without ever having quantum communication between them. Another one is that the QSVD can be turned into an encoder of quantum information.

Indeed, this proposal is a variational quantum algorithm, much in the spirit of recent developments in the field of quantum computation for the noisy intermediate-scale quantum era [Pre18]. This, as discussed in previous chapters, means that the basic circuits may be shallow (accuracy can be increased by increasing the depth) and therefore amenable to implementation on near-term quantum computers without error correction.

[Pre18] Preskill, "Quantum Computing in the NISQ era and beyond"

4.1 CLASSICAL SINGULAR VALUE DECOMPOSITION

The Singular Value Decomposition is a powerful mathematical technique ubiquitously used to analyze tensors with two indices. It simply says that any such tensor can be cast onto a diagonal form using two unitary matrices that act on each of its indices.

Let us briefly review how the SVD is computed. Consider a bipartite pure state $|\psi\rangle_{AB} \in \mathcal{H}_A \otimes \mathcal{H}_B$,

$$|\psi\rangle_{AB} = \sum_{i=1}^{d_A} \sum_{j=1}^{d_B} c_{ij}\, |e_i\rangle_A |e_j\rangle_B\,, \tag{4.1}$$

where $d_{A,B}$ are the dimensions of the susbsystems Hilbert spaces $\mathcal{H}_{A,B}$, $\{|e_k\rangle_{A,B}\}$ are the computational-basis states in $\mathcal{H}_{A,B}$, and the complex coefficients c_{ij} obey a normalization relation. This state can be written in its Schmidt form,

$$|\psi\rangle_{AB} = \sum_{i=1}^{\chi} \lambda_i\, |u_i\rangle_A |v_i\rangle_B\,, \tag{4.2}$$

where χ is the Schmidt rank (*i.e.* the number of Schmidt coefficients different from zero), which is always equal or smaller than the minimum of d_A and d_B; λ_i are real positive eigenvalues that can be sorted in decreasing order, and $\{|u_i\rangle_A\}$ and $\{|v_i\rangle_B\}$ form a orthonormal basis for subsystems A and B respectively.

The analytical way to find the SVD of a given vector as in Eq. 4.1 needs to start from the tensor c_{ij}, then compute the reduced density matrix for each subsystem, $\rho_A = Tr_B|\psi\rangle_{AB}\langle\psi|$ and $\rho_B = Tr_A|\psi\rangle_{AB}\langle\psi|$, and then perform two diagonalizations, $\rho_A = \sum_{i=1}^{\chi} \lambda_i^2 |u_i\rangle_A\langle u_i|$ and $\rho_B = \sum_{i=1}^{\chi} \lambda_i^2 |v_i\rangle_B\langle v_i|$. As a result, the original vector can be cast in the basis of the eigenvectors $\{|u_i\rangle_A\}$ and $\{|v_i\rangle_B\}$ of both diagonalizations that share the same eigenvalues. The sign of each λ_i can be taken positive as a phase can always be absorbed into either $|u_i\rangle_A$ or $|v_i\rangle_B$.

Note that the Schmidt rank χ is in itself a first measure of entanglement. Furthermore, the usefulness of the SVD can be illustrated by computing the von Neumann entropy S of this state for the A-B bipartition: $S = -\mathrm{Tr}(\rho_A \log \rho_A) = -\mathrm{Tr}(\rho_B \log \rho_B) = -\sum_{i=1}^{\chi} \lambda_i^2 \log \lambda_i^2$. It also follows that all Rényi entropies can be computed once the eigenvalues of the SVD are known [Rén61].

[Rén61] Rényi, *On measures of entropy and information*

The classical construction of the SVD can only be used on an actual quantum state after performing its full tomography. Indeed, for a state made out of n qubits it is necessary to perform measurements in 3^n different settings (*i.e.* one for each non-commuting combination of tensor products of Pauli operators $\{\sigma_x, \sigma_y, \sigma_z\}$) to reconstruct the original tensor c_{ij} [MRL08], and then perform all the computations sketched above. Furthermore, the classical computation of the entropy to be performed from the tensor describing the state may get out of reach for large systems since it scales exponentially with the number of qubits.

[MRL08] Mohseni et al., “Quantum-process tomography: Resource analysis of different strategies”

4.2 CIRCUIT FOR THE QUANTUM SINGULAR VALUE DECOMPOSITION

In this section, we present a novel way to compute the eigenvalues and obtain the physical eigenvectors of the SVD of a pure state $|\psi\rangle_{AB}$ using a quantum circuit that we shall name QSVD for Quantum Singular Value Decomposer. Our technique needs copies of the original state.

The key idea of our method is to find a circuit that provides the following transformation of the original state:

$$U_A \otimes V_B\ |\psi\rangle_{AB} = \sum_{i=1}^{\chi} \lambda_i\, e^{i\alpha_i} |e_i\rangle_A |e_i\rangle_B\,, \tag{4.3}$$

where $U_A|u_i\rangle_A = e^{i\beta_i}|e_i\rangle_A$ and $V_B|v_i\rangle_B = e^{i\gamma_i}|e_i\rangle_B$, with $\alpha_i = \beta_i + \gamma_i \in [0, 2\pi)$; the i in the exponent is the imaginary unit.

The way to find the desired circuit emerges from the following observation. Given that the new Schmidt bases for the two subsystems are right away the computational-basis vectors (up to individual global phases), each time we perform a measurement, we should find *exact output coincidence* between the respective observations in A and B (Figure 4.1). Let us consider the example of two subspaces of two qubits. Then, if the result in the first subsystem turns out to be *e.g.* 00, the result in the second subsystem should also be 00. There is always a pair of unitaries U_A and V_B achieving this exact output coincidence since they simply correspond to a change of basis from the Schmidt eigenvectors.

Note that the entanglement spectrum between A and B (*i.e.* the eigenvalues $\{\lambda_i\}$ of the reduced density matrices ρ_A, ρ_B) has not changed, nor has the von Neumann entropy. This allows to obtain an estimation of the Schmidt coefficients $\{\lambda_i\}$, which will just be the observed, normalized probabilities for each possible *coincident* outcome for the two subsystems, obtained from repeated preparation of the state, application of the QSVD and measurement.

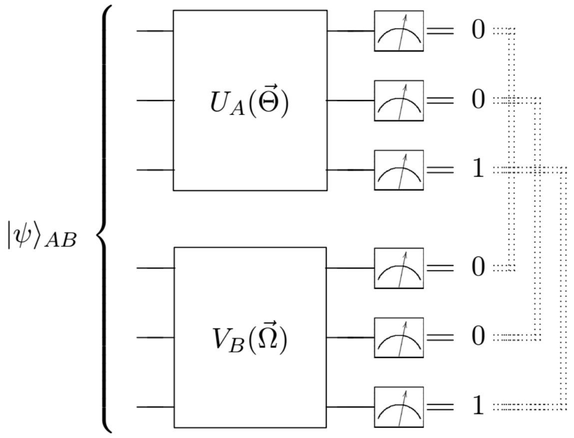

FIGURE 4.1: Parametrized unitary transformations implementing the Quantum Singular Value Decomposer (QSVD). Training is based on demanding exact output coincidence for both parties and for every measurement.

In turn, these coefficients provide several entanglement figures of merit, such as the von Neumann entropy.

Once the two unitaries U_A and V_B have been obtained, it is now possible to reconstruct the vectors that would be needed in the original SVD, up to a complex phase. They simply correspond to

$$e^{-i\beta_i}|u_i\rangle_A = U_A^\dagger\,|e_i\rangle_A\,, \qquad e^{-i\gamma_i}|v_i\rangle_B = V_B^\dagger\,|e_i\rangle_B\,, \tag{4.4}$$

where $|e_i\rangle$ can be created by applying X-gates onto the initial $|00...00\rangle$ state at the beginning of the computation. The algorithm has taken the original state to a very specific form, the one of exact output coincidence, to determine the unitaries, and then the adjoint of the same unitaries are used to reconstruct the eigenvectors. The global phases $\{\beta_i, \gamma_i\}$ are irrelevant in the characterization of the individual eigenvectors, but if one is interested in the relative phases between these vectors in the original $|\psi\rangle_{AB}$, then one may need to resort to tomography.

The algorithm we have presented has some extra degrees of freedom. For the sake of clarity, we explicitly demanded exact output coincidence. Nevertheless, this is not necessary in a strict sense. It suffices that each unique output from subsystem A is matched by some other unique output from subsystem B. This means that there is freedom of permutation for, say, output from B. Such a permutation is just another unitary on the B side. However, freedom of permutation (*i.e.* alternative training) must be avoided in two further applications of the QSVD (SWAP without quantum communication and quantum encoder), which we present below. Freedom of phase, in contrast, does not have any effect on them.

Another obvious comment of the algorithm is related to the possibility of having partitions with different dimensions. In such a case, the larger subsystem will have some irrelevant elements in its basis that will never tick on measurement.

4.3 VARIATIONAL QUANTUM SINGULAR VALUE DECOMPOSER

The key role of the exact output coincidence is the guide to construct a quantum circuit to perform this task. Indeed, it is possible to train a variational version of the QSVD that will approximate the exact QSVD.

We first need to construct the two needed unitaries as a quantum circuit made of entangling gates and single qubit rotations. This circuit is thus characterized by a set of classical parameters. We may choose for instance the architecture shown in Figure 4.2, where all the free parameters correspond to angles of rotation for single qubits $\vec{\Theta}$ and $\vec{\Omega}$ for subsystem A and B respectively. The variational form of the QSVD reads now

$$|\psi\rangle_{AB} \xrightarrow{QSVD} U_A(\vec{\Theta}) \otimes V_B(\vec{\Omega})\, |\psi\rangle_{AB} = \sum_{i=1}^{\chi} \lambda_i\, e^{i\alpha_i} |e_i\rangle_A |e_i\rangle_B\,. \tag{4.5}$$

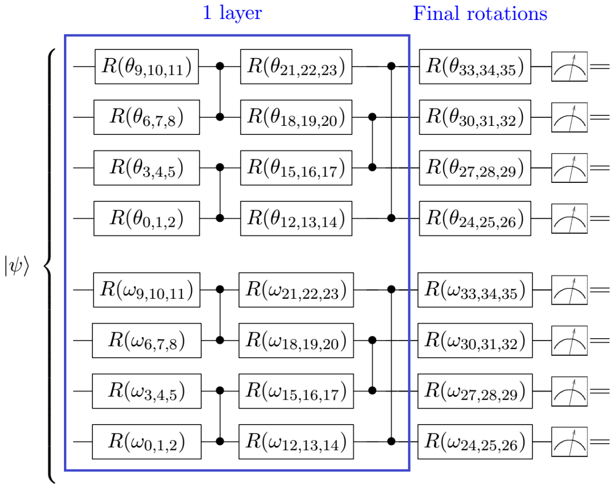

FIGURE 4.2: Architecture of the variational circuit employed in the QSVD. Several layers of gates are applied consecutively in order to improve accuracy, and the circuit always ends with a final set of single-qubit rotations prior to measurement. The notation stands for $R(\theta_{\alpha,\beta,\gamma}) \equiv Rz(\theta_\alpha)Rx(\theta_\beta)Rz(\theta_\gamma)$. If the number of qubits is odd in a given subsystem, then an extra CZ between the first and last qubit of the subsystem is added after each complete rotation.

At the outset, random values for the parameters might be used, and the circuit does not issue states that show exact output coincidence for all measurements. A figure of merit for the wrong answer is simply the total amount of non-coincidental measurement outcomes, which shall be minimized. In order to help accelerate convergence, different outcomes for each subsystem may be penalized by their Hamming distance, which is just the number of symbols that are different in the binary representation of the two results. Thus, the cost function C to be minimized simply is

$$C \equiv \sum_j d_H(M_j^A, M_j^B) \equiv \sum_q \frac{1 - \langle \sigma_z^{q,A} \sigma_z^{q,B} \rangle}{2}\,, \tag{4.6}$$

where d_H denotes the Hamming distance and $M_j^{A,B}$ are the results of the j-th measurement in the computational basis for A and B, respectively. Equivalently, it can also be defined in terms of 2-local σ_z Pauli operators, where the index q runs over all the qubits in the smallest subsystem. We may now apply machine learning techniques to find the optimal parameters that provide exact output coincidence. Notice that this cost function has a value of zero if and only if the Singular Value Decomposition is successfully completed. Note as well that it is defined in terms of 2-local observables, and therefore, it does not suffer, for shallow quantum circuits, from the problem of exponentially vanishing gradients [Cer+21b]. We emphasize that there is no need to perform any tomography nor to involve any measurement of non-trivial observables. This simplification is related to the fact that there is no need to measure any relative phase. Therefore, the QSVD implies an exponential reduction in the number of measurement settings compared to state tomography, which requires 3^n.

[Cer+21b] Cerezo et al., "Cost function dependent barren plateaus in shallow parametrized quantum circuits"

The convergence of the method depends on two distinct elements. First, the potential convergence of a variational QSVD to the exact QSVD is controlled by the Solovay-Kitaev theorem [Kit+02]. Indeed, we are just looking for an approximation to a unitary using a complete set of gates. This means that there exists a quantum circuit that approximates the desired unitary with error δ, *i.e.*, $|U_{exact} - U(\vec{\Theta})| < \delta$, with a number of gates k that scales as $k \sim \log^c \frac{1}{\delta}$, with $1 \leq c < 4$, for a fixed number of qubits [HRC02]. In other words, the error in the unitary may potentially decrease exponentially with the depth of the circuit for a fixed number of qubits. This, in practice, will depend on the circuit ansatz and the success of the optimization procedure. The number of layers of the variational circuit has to increase polynomially with the system size. Under these conditions, a classical search algorithm needs only to explore a polynomial number of dimensions. Finding the optimal parameters may nonetheless encounter exponentially vanishing gradients [McC+18] or local minima for deep quantum circuits, that need to be circumvented using appropriate optimization strategies [WHT15; GSL18; Mol+18; Gra+19; Küb+20].

[Kit+02] Kitaev et al., *Classical and Quantum Computation*

[HRC02] Harrow et al., "Efficient discrete approximations of quantum gates"

[McC+18] McClean et al., "Barren plateaus in quantum neural network training landscapes"

[WHT15] Wecker et al., "Progress towards practical quantum variational algorithms"

[GSL18] Garcia-Saez and Latorre, "Addressing hard classical problems with adiabatically assisted variational quantum eigensolvers"

[Mol+18] Moll et al., "Quantum optimization using variational algorithms on near-term quantum devices"

[Gra+19] Grant et al., "An initialization strategy for addressing barren plateaus in parametrized quantum circuits"

[Küb+20] Kübler et al., "An adaptive optimizer for measurement-frugal variational algorithms"

Second, the QSVD samples from a multinomial distribution. As such, the additive error for each output probability p_i scales as $\sqrt{p_i(1-p_i)/s}$, where s is the number of samples. The total number of measurements is related to the error which is aimed at, which in turn will depend on the Schmidt rank. We may then consider two different cases: (i) the Schmidt rank increases polynomially with the number of qubits, and (ii) the Schmidt rank increases exponentially with the number of qubits. In the first case (i), only a poly(n) number of measurements is needed to achieve a low relative error, whereas, in a worst-case scenario (ii), this number is exponential if one is to estimate all eigenvalues with a low relative error. The latter case follows naturally from the fact that we are asking for an exponential amount of information. However, many physically relevant states, *e.g.* in condensed matter systems, do not exhibit an exponentially large Schmidt rank [Ami+08].

[Ami+08] Amico et al., "Entanglement in many-body systems"

The variational approach to the QSVD can be verified on a simulation. For instance, we have considered random states with $c_{ij} = a_{ij} + i\, b_{ij}$ such that a_{ij} and b_{ij} are random real numbers between -0.5 and 0.5, further restricted by a global normalization. These states tend to have very large entanglement [BŻ17]. We have simulated states with a total number of 6 qubits and natural bipartition, disregarding the presence of experimental noise and the impact of finite sampling. We have analyzed 500 instances for the 1 and 2 layers case and 200 instances for the 3, 4, and 5 layers case. The

[BŻ17] Bengtsson and Życzkowski, *Geometry of quantum states: an introduction to quantum entanglement* classical method employed in the optimization loop was L-BFGS-B, which is gradient-based and involves the estimation of the inverse Hessian matrix. We utilized the implemented version of the open-source Python package SciPy Optimize [Vir+20]. The mean number of optimization steps is of the order of a few hundred.

[Vir+20] Virtanen et al., “SciPy 1.0: fundamental algorithms for scientific computing in Python”

The basic unit-cell or layer of the variational ansatz employed in the simulations is shown in Figure 4.2. These layers are used as buildings blocks to construct deeper circuits by consecutively applying the architecture of the single layer, followed by a final set of single-qubit rotations. The number of layers on a circuit controls the accuracy of the estimation, as previously discussed. One layer has depth 8 (10 if the number of qubits of a subsystem is odd), so the depth of the circuit as a function of the number l of layers is $8l+3$ $(10l+3)$. The total number of 1-qubit gates is $6ln+3n$, where n is the number of qubits, and that of 2-qubit gates is ln $(ln+2l)$. Therefore, depth and number of gates are efficient in both the number of qubits and the number of layers.

Figure 4.3 left shows the entanglement entropy computed from the trained QSVD circuit *vs.* the exact entropy. As suggested by the Solovay-Kitaev theorem, we observe fast convergence of results for every instance we have analyzed (Figure 4.3 right). The variational circuit approaches the exact result as we increase the number of layers, whatever the entanglement is. In this respect, it is worth mentioning that we have also analyzed Absolute Maximally Entangled (AME) states [CLLG19], for which the convergence of the variational QSVD is fast and faithful.

[CLLG19] Cervera-Lierta et al., “Quantum circuits for maximally entangled states”

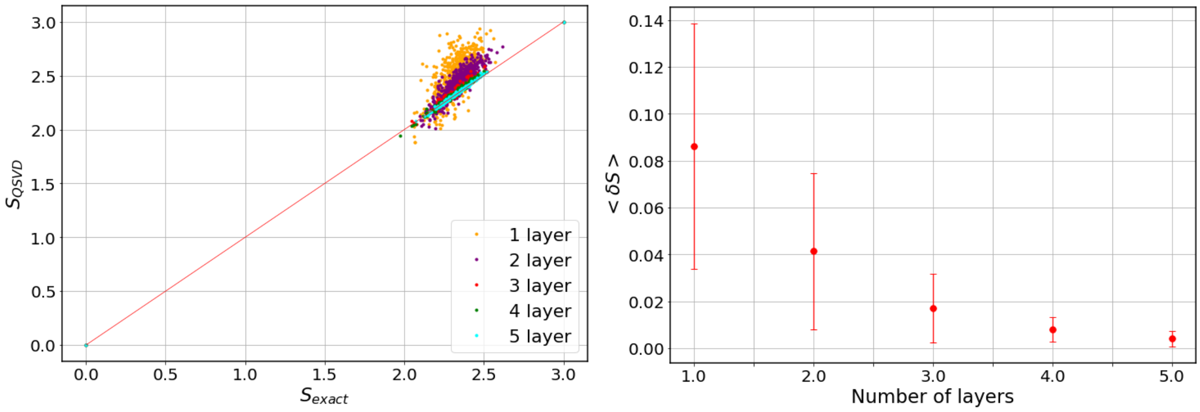

FIGURE 4.3: *Left:* Von Neumann entropy computed from the variational form of QSVD *vs.* exact entropy, for random states (including a product state and AME state) of 6 qubits and natural bipartition. As the number of layers increases, we observe convergence towards the exact entropy. *Right:* Mean relative error in the estimation of the entropy *vs.* number of layers (error bars represent the standard deviation). The error decreases exponentially with the depth of the circuit, as suggested by the Solovay-Kitaev theorem.

4.4 ADDITIONAL APPLICATIONS

4.4.1 *SWAP with classical communication*

A peculiar spinoff of the QSVD circuit is the possibility of performing a SWAP operation between parties A and B without using any gate that connects both subsystems. This is in contrast with the standard SWAP, where each pair of swapped qubits would need a series of CNOT gates.

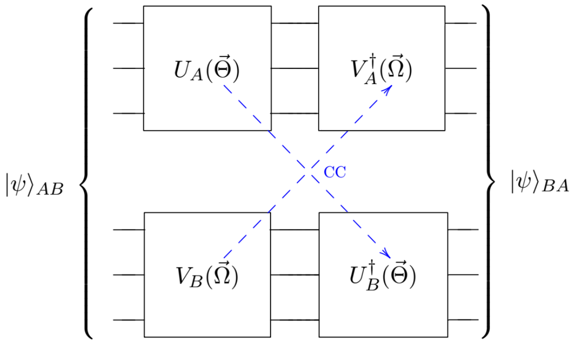

FIGURE 4.4: Application of the QSVD followed by the adjoint $U^\dagger$ and $V^\dagger$ gates acting on opposite subsystems, mediated by classical communication (CC) of the optimal parameters, allows to perform a long-distance SWAP operation without the need of any quantum communication between subsystems.

The idea is shown in Figure 4.4. It is enough to apply the QSVD to $|\psi\rangle_{AB}$, then apply the adjoint $U^\dagger$ and $V^\dagger$ gates but acting on the opposite subsystem. That is:

$$(V_A^\dagger \otimes U_B^\dagger)(U_A \otimes V_B)|\psi\rangle_{AB} = |\psi\rangle_{BA}. \tag{4.7}$$

The implementation of the adjoint unitaries only needs classical communication between parties A and B since each unitary is characterized by a set of classical parameters. Notice as well that none of the gates will ever cross the barrier between the two systems.

Performing a SWAP without quantum communication opens the possibility of swapping at long distances. Let us imagine Alice and Bob received their pieces of a given common state. They can then measure their copies and notify their results publicly. Without further communication, they can improve their variational QSVD. After several iterations, they will observe exact coincidence. Both parties can then communicate the characterization of their respective unitaries classically and run once more the QSVD adding the adjoint exchanged gates at the end. They will then have achieved a long-distance SWAP without quantum interaction. Of course, standard SWAP is much more powerful as it acts on a single copy of any unknown state. The price to be paid is the need for entangling gates across the two subsystems.

4.4.2 *Quantum singular value decomposer as an encoder*

The QSVD algorithm has a further spinoff. Let us consider, for the sake of simplicity, a system of n qubits where we apply the QSVD algorithm to a given bipartition. If we consider the final state of the circuit, the exact coincidence of the parties can be used to set to $|0\rangle$ all the qubits of *e.g.* party A. It is only needed to apply a CNOT between each pair of coincident qubits controlled at A and targeted to B, as illustrated in Figure 4.5. The QSVD plus a series of CNOTs corresponds to a quantum encoder designed to compress the initial state onto $(CNOT_1 \dots CNOT_{n/2})(U_A \otimes V_B)|\psi\rangle_{AB} = |00\dots0\rangle_A|\phi\rangle_B$, where $|\phi\rangle_B = \sum_{i=1}^{\chi} \lambda_i\, e^{i\alpha_i}|e_i\rangle_B$. All the information of the original state has been packed into one subsystem. The circuit being unitary, this encoding can be exactly decoded back onto the total system.

The same idea can be reversed. Let us imagine that we are interested in creating a random state that displays a very precise entanglement structure.

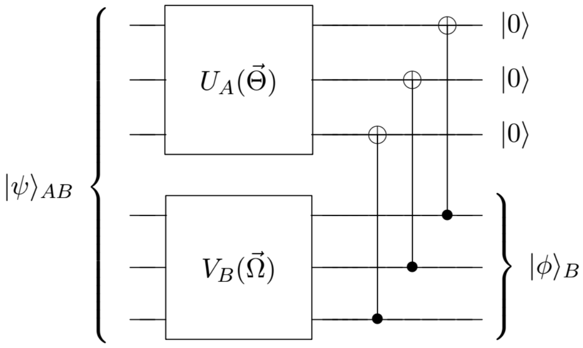

FIGURE 4.5: Further use of CNOT gates makes QSVD an encoder of the original quantum state $|\psi\rangle_{AB}$ onto one of its parts $|\phi\rangle_B$.

The procedure would be to first manage to create the following superposition on a subsystem $|\psi\rangle_A = \sum_i \lambda_i e^{i\alpha_i}|e_i\rangle_A$. Then a series of CNOTs connecting each qubit to an ancilla would lead to $|\psi\rangle_{AB} = \sum_i \lambda_i e^{i\alpha_i}|e_i\rangle_A|e_i\rangle_B$. Finally, the state can be randomized by taking arbitrary unitaries on A and B.

4.5 OUTLOOK

We have presented a novel algorithm, QSVD, that provides Schmidt eigenvalues and eigenvectors of any bipartite pure state, given many copies of it. Its key idea can be traced to demand *exact output coincidence* on any measurement of the two parties that make the system.

The QSVD can be used to analyze the entanglement which is present in the result of any algorithm. For instance, if a variational quantum circuit is trained to minimize the energy of *e.g.* the Ising model, the final result can be run with the addition of the variational form of QSVD. The results would then allow to check the logarithmic growth of the entropy at criticality.

The QSVD seems to be a natural structure to achieve a number of quantum tasks. Here we have analyzed the possibility to achieve a SWAP operation without any gate that connects qubits from both sides of the state. We have also shown that QSVD plus a series of CNOTs is tantamount to a quantum encoder.

Finally, let us mention that after the realization of this work, other variational quantum algorithms have been proposed. On the one hand, Ref. [Cer+20a] addresses the general case when the quantum states are described as density matrices. On the other hand, in Ref. [WSW21], the authors propose a different loss function inspired by properties of singular values.

[Cer+20a] Cerezo et al., "Variational Quantum State Eigensolver"

[WSW21] Wang et al., "Variational quantum singular value decomposition"

5

Solving linear systems of equations

> "*It does not say in the Bible that all laws of nature are expressible linearly!*"
> —Enrico Fermi

Linear systems of equations play an important role in many areas of science and technology, including machine learning [Bis06; Alp10], solving partial differential equations [Eva10], fitting polynomial curves [Bre95], and analyzing electrical circuits [SS11]. In the past decade, significant attention has been given to the possibility of solving linear systems on quantum computers. Classically solving an $N \times N$ linear system (N equations for N unknowns) scales polynomially in N. In contrast, Harrow-Hassidim-Lloyd (HHL) introduced a quantum algorithm that scales logarithmically in N, suggesting that quantum computers may provide an exponential speedup for certain linear system problems [HHL09]. More precisely, the HHL algorithm treats the Quantum Linear Systems Problem (QLSP), where the goal is to prepare a quantum state $|x\rangle$ that is proportional to a vector $\vec{x}$ that satisfies the equation $A\vec{x} = \vec{b}$. If both A and $\vec{b}$ are sparse, then for a fixed precision ϵ in the solution, the complexity of HHL scales polynomially in $\log N$ and κ, where κ is the condition number of A, *i.e.*, the ratio of the largest to the smallest singular value. Further improvements to HHL have reduced the complexity to linear κ scaling [Amb10; SSO19] and polylogarithmic scaling in $1/\epsilon$ [CKS17; CGJ18], as well as improved the sparsity requirements [WZP18].

[Bis06] Bishop, *Pattern Recognition and Machine Learning*

[Alp10] Alpaydin, *Introduction to Machine Learning*

[Eva10] Evans, *Partial differential equations*

[Bre95] Bretscher, *Linear Algebra With Applications*

[SS11] Spielman and Srivastava, "Graph Sparsification by Effective Resistances"

[HHL09] Harrow et al., "Quantum algorithm for linear systems of equations"

[Amb10] Ambainis, "Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations"

[SSO19] Subaşı et al., "Quantum Algorithms for Systems of Linear Equations Inspired by Adiabatic Quantum Computing"

[CKS17] Childs et al., "Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision"

[CGJ18] Chakraborty et al., "The power of block-encoded matrix powers: improved regression techniques via faster Hamiltonian simulation"

[WZP18] Wossnig et al., "Quantum Linear System Algorithm for Dense Matrices"

The aforementioned quantum algorithms hold promise for the future when large-scale quantum computers exist with enough qubits for quantum error correction. The timescale for such computers remains an open question but is typically estimated to be on the order of two decades. On the other hand, commercial quantum computers currently exist with ~ 50 noisy qubits, with the number of qubits rapidly increasing. A crucial question is how to make use of such noisy intermediate-scale quantum (NISQ) computers [Pre18]. In principle, one can implement the aforementioned quantum algorithms on NISQ devices; however, noise limits the problem size to be extremely small. For example, the HHL algorithm has been implemented with superconducting qubits [Zhe+17; LJL19], nuclear magnetic resonance (NMR) [Pan+14], and photonic devices [Cai+13; Bar+14], but these experiments were limited to a problem size of 2×2. More recently, an alternative approach based on an adiabatic-inspired quantum algorithm [SSO19] was implemented with NMR for an 8×8 problem, and this appears to be the current record for the largest linear system solved with a gate-based quantum computer [Wen+19].

[SSO19] Subaşı et al., "Quantum Algorithms for Systems of Linear Equations Inspired by Adiabatic Quantum Computing"

As discussed in previous chapters, an interesting strategy to make use of NISQ devices is to employ variational quantum algorithms (VQAs). Recall that VQAs manage to reduce quantum circuit depth at the expense of additional classical optimization. Specifically, VQAs employ a short-depth quantum circuit to efficiently evaluate a cost function, which depends on the parameters of a quantum gate sequence and then leverage well-established classical optimizers to minimize this cost function. For example, while Shor's algorithm for factoring is not a near-term algorithm, recently, a VQA for factoring was introduced, potentially making factoring nearer term [Ans+19]. Other VQAs have been proposed for chemistry [Per+14; Cao+19; HWB19; Jon+19; Pra+21], simulation [LB17; Kok+19; Hey+19; Yua+19; Cî+20; BVC21; Tan+22], compiling [Kha+19b; JB22], quantum foundations [Arr+19], anomaly detection [Kot+21], error correction [Joh+17; LCM22], fidelity estimation [Cer+20b; Che+21], and metrology [Koc+20].

[Ans+19] Anschuetz et al., "Variational quantum factoring"

[Kha+19b] Khatri et al., "Quantum-assisted quantum compiling"

[JB22] Jones and Benjamin, "Robust quantum compilation and circuit optimisation via energy minimisation"

[Arr+19] Arrasmith et al., "Variational consistent histories as a hybrid algorithm for quantum foundations"

[Kot+21] Kottmann et al., "Variational quantum anomaly detection: Unsupervised mapping of phase diagrams on a physical quantum computer"

[Joh+17] Johnson et al., "QVECTOR: an algorithm for device-tailored quantum error correction"

[LCM22] Locher et al., "Quantum Error Correction with Quantum Autoencoders"

[Cer+20b] Cerezo et al., "Variational quantum fidelity estimation"

[Che+21] Chen et al., "Variational quantum algorithms for trace distance and fidelity estimation"

[Koc+20] Koczor et al., "Variational-state quantum metrology"

In this chapter, we propose a VQA for solving the QLSP. This algorithm, called the Variational Quantum Linear Solver (VQLS), employs a cost function that quantifies either the global or local closeness of the quantum states $A|x\rangle$ and $|b\rangle$, where the latter is a normalized version of $\vec{b}$. We also provide efficient quantum circuits to estimate the cost functions. Furthermore, we derive operational meaning for our cost functions, as upper bounds on ϵ^2/κ^2. This is crucial since it gives a termination criterion for VQLS that guarantees a desired precision ϵ.

It is important to emphasize that all VQAs are heuristic algorithms, making rigorous complexity analysis difficult. Nevertheless, our numerical simulations (without finite sampling, and both for specific A and for randomly chosen A) indicate that the run time of VQLS scales efficiently in κ, ϵ, and N. Namely, we find evidence of (at worst) linear scaling in κ, logarithmic scaling in $1/\epsilon$, and polylogarithmic scaling in N.

We employ Rigetti's Quantum Cloud Services[1] to implement VQLS. With their quantum hardware, we were able to successfully solve a particular linear system of size 1024 × 1024. We are therefore optimistic that VQLS could provide a near-term approach to the QLSP.

5.1 VARIATIONAL QUANTUM LINEAR SOLVER

Figure 5.1 shows a schematic diagram of the VQLS algorithm. The input to VQLS is: (1) an efficient gate sequence U that prepares a quantum state $|b\rangle$ that is proportional to the vector $\vec{b}$, and (2) a decomposition of the matrix A into a linear combination of L unitaries of the form

$$A = \sum_{l=1}^{L} c_l A_l\,, \tag{5.1}$$

where the A_l are unitaries, and the c_l are complex numbers. The assumption that A is given in this form is analogous to the assumption that the Hamiltonian H in the variational quantum eigensolver [Per+14] is given as a linear combination of Pauli operators $H = \sum_{l=1}^{L} c_l \sigma_l$, where naturally one makes the assumption that L is only a polynomial function of the number of qubits,

[Per+14] Peruzzo et al., "A variational eigenvalue solver on a photonic quantum processor"

---

[1] Rigetti's cloud access through Amazon Web Services, Feb. 2022.

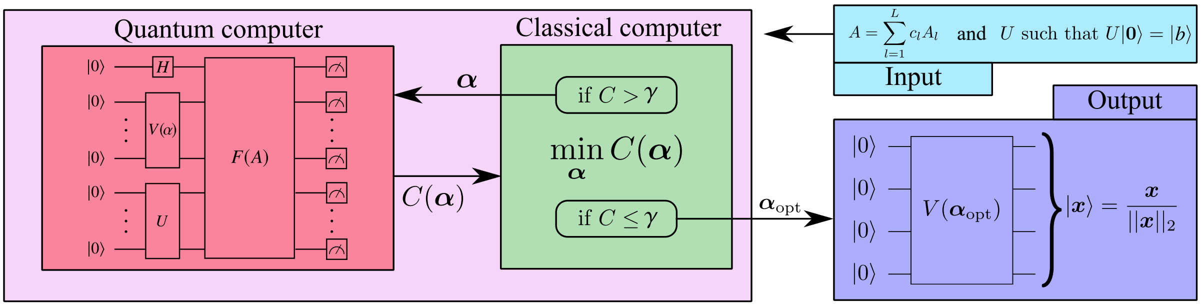

FIGURE 5.1: Schematic diagram for the VQLS algorithm. The input to VQLS is a matrix A written as a linear combination of unitaries A_l and a short-depth quantum circuit U which prepares the state $|b\rangle$. The output of VQLS is a quantum state $|x\rangle$ that is approximately proportional to the solution of the linear system $A\vec{x} = \vec{b}$. Parameters $\vec{\alpha}$ in the ansatz $V(\vec{\alpha})$ are adjusted in a hybrid quantum-classical optimization loop until the cost $C(\vec{\alpha})$ (local or global) is below a user-specified threshold. When this loop terminates, the resulting gate sequence $V(\vec{\alpha}_{\text{opt}})$ prepares the state $|x\rangle = \vec{x}/||\vec{x}||_2$, from which observable quantities can be computed. Furthermore, the final value of the cost $C(\vec{\alpha}_{\text{opt}})$ provides an upper bound on the deviation of observables measured on $|x\rangle$ from observables measured on the exact solution.

n. Additionally, we assume $\kappa < \infty$ and $\|A\| \leq 1$, and that the A_l unitaries can be implemented with efficient quantum circuits.

With this input, the Quantum Linear Systems Problem (QLSP) is to prepare a state $|x\rangle$ such that $A|x\rangle$ is proportional to $|b\rangle$. To solve this problem, VQLS employs an ansatz for the gate sequence $V(\vec{\alpha})$ that prepares a potential solution $|x(\vec{\alpha})\rangle = V(\vec{\alpha})|\vec{0}\rangle$. The parameters $\vec{\alpha}$ are input to a quantum computer, which prepares $|x(\vec{\alpha})\rangle$ and runs an efficient quantum circuit that estimates a cost function $C(\vec{\alpha})$. The precise details of the cost function and its estimation are discussed below. We simply remark here that $C(\vec{\alpha})$ quantifies how much component $A|x\rangle$ has orthogonal to $|b\rangle$. The value of $C(\vec{\alpha})$ from the quantum computer is returned to the classical computer which then adjusts $\vec{\alpha}$ (via a classical optimization algorithm) in an attempt to reduce the cost. This process is iterated many times until one reaches a termination condition of the form $C(\vec{\alpha}) \leq \gamma$, at which point we say that $\vec{\alpha} = \vec{\alpha}_{\text{opt}}$.

VQLS outputs the parameters $\vec{\alpha}_{\text{opt}}$, which can then be used to prepare the quantum state $|x(\vec{\alpha}_{\text{opt}})\rangle = V(\vec{\alpha}_{\text{opt}})|\vec{0}\rangle$. One can then measure observables of interest on the state $|x(\vec{\alpha}_{\text{opt}})\rangle$ in order to characterize the solution vector. Due to the operational meaning of our cost function (discussed below), one can upper bound the deviation of observable expectation values for $|x(\vec{\alpha}_{\text{opt}})\rangle$ from those of the true solution, based on the value of the cost function. Hence, before running VQLS, one can decide on a desired error tolerance ϵ, where

$$\epsilon = (1/2)\operatorname{Tr}\left|\, |x_0\rangle\langle x_0| - |x(\vec{\alpha}_{\text{opt}})\rangle\langle x(\vec{\alpha}_{\text{opt}})|\, \right| \qquad (5.2)$$

is the trace distance between the exact solution $|x_0\rangle$ and the approximate solution $|x(\vec{\alpha}_{\text{opt}})\rangle$. This ϵ then translates into a threshold value γ that the final cost $C(\vec{\alpha}_{\text{opt}})$ must achieve (see Eq. 5.10 for the relation between ϵ and γ).

5.1.1 *Cost functions*

For simplicity, we write $|x(\vec{\alpha})\rangle$ as $|x\rangle$ henceforth. Here we discuss several reasonable cost functions. A simple, intuitive cost function involves the overlap between the (unnormalized) projector $|\psi\rangle\langle\psi|$, with $|\psi\rangle = A|x\rangle$, and the subspace orthogonal to $|b\rangle$, as follows:

$$\widehat{C}_G = \mathrm{Tr}(|\psi\rangle\langle\psi|(\mathbb{1} - |b\rangle\langle b|)) = \langle x|H_G|x\rangle\,. \tag{5.3}$$

We note that one can view this cost function as the expectation value of an effective Hamiltonian

$$H_G = A^\dagger(\mathbb{1} - |b\rangle\langle b|)A, \tag{5.4}$$

[SSO19] Subaşı et al., "Quantum Algorithms for Systems of Linear Equations Inspired by Adiabatic Quantum Computing"

which is similar to the final Hamiltonian in Ref. [SSO19]. The $\widehat{C}_G$ function is small if $|\psi\rangle$ is nearly proportional to $|b\rangle$ or if the norm of $|\psi\rangle$ is small. The latter does not represent a true solution, and hence to deal with this, one can divide $\widehat{C}_G$ by the norm of $|\psi\rangle$ to obtain

$$C_G = \widehat{C}_G/\langle\psi|\psi\rangle = 1 - |\langle b|\Psi\rangle|^2\,, \tag{5.5}$$

where $|\Psi\rangle = |\psi\rangle\,/\sqrt{\langle\psi|\psi\rangle}$ is a normalized state.

[Cer+21b] Cerezo et al., "Cost function dependent barren plateaus in shallow parametrized quantum circuits"

We emphasize that global cost functions such as those in Eqs. 5.3 and 5.5 can exhibit barren plateaus, *i.e.*, cost function gradients that vanish exponentially in the number of qubits n, see Ref. [Cer+21b]. To improve trainability for large n, one can introduce local versions of these costs, as follows:

$$\widehat{C}_L = \langle x|H_L|x\rangle\,, \qquad C_L = \widehat{C}_L/\langle\psi|\psi\rangle\,, \tag{5.6}$$

where the effective Hamiltonian is

$$H_L = A^\dagger U\left(\mathbb{1} - \frac{1}{n}\sum_{j=1}^{n}|0_j\rangle\langle 0_j| \otimes \mathbb{1}_{\bar{j}}\right)U^\dagger A, \tag{5.7}$$

with $|0_j\rangle$ the zero state on qubit j and $\mathbb{1}_{\bar{j}}$ the identity on all qubits except qubit j. One can show that

$$\widehat{C}_L \le \widehat{C}_G \le n\widehat{C}_L\,, \qquad C_L \le C_G \le nC_L\,, \tag{5.8}$$

which implies that $\widehat{C}_L = 0 \leftrightarrow \widehat{C}_G = 0$ and $C_L = 0 \leftrightarrow C_G = 0$. We assume that κ is not infinite (*i.e.*, that A is full rank) and hence that $\langle\psi|\psi\rangle \neq 0$. This implies that all four cost functions vanish under precisely the same conditions, namely, when $|\psi\rangle \propto |b\rangle$, which is the case when $|x\rangle$ is a solution to the QLSP.

Let us provide a proof for Eq. 5.8. For the lower bound, let $\Pi_G = \mathbb{1} - |\vec{0}\rangle\langle\vec{0}|$ and $\Pi_L = \mathbb{1} - \frac{1}{n}\sum_{j=1}^{n}|0_j\rangle\langle 0_j| \otimes \mathbb{1}_{\bar{j}}$. Using the fact that $|0_j\rangle\langle 0_j| \otimes \mathbb{1}_{\bar{j}} \ge |\vec{0}\rangle\langle\vec{0}|$, we have $\Pi_G \ge \Pi_L$ and hence $H_G \ge H_L$. This implies that $\widehat{C}_G \ge \widehat{C}_L$ and $C_G \ge C_L$.

For the upper bound, note that $\Pi_G = \sum_{\vec{z}\neq\vec{0}}|\vec{z}\rangle\langle\vec{z}|$. Let $\mathcal{S}_j$ denote the set of all bitstrings that have a one at position j, and let $\mathcal{S} = \bigcup_j \mathcal{S}_j$ denote the union of all of these sets. Then

$$n\Pi_L = \sum_j \sum_{\vec{z}\in\mathcal{S}_j}|\vec{z}\rangle\langle\vec{z}| \ge \sum_{\vec{z}\in\mathcal{S}}|\vec{z}\rangle\langle\vec{z}| = \sum_{\vec{z}\neq\vec{0}}|\vec{z}\rangle\langle\vec{z}| = \Pi_G\,. \tag{5.9}$$

Hence we have $nH_L \ge H_G$, which implies $n\widehat{C}_L \ge \widehat{C}_G$ and $nC_L \ge C_G$.

Equation 5.8 implies the faithfulness of the cost functions as follows. Because $\Pi_G \ge 0$ and $\Pi_L \ge 0$, we have that all four cost functions are

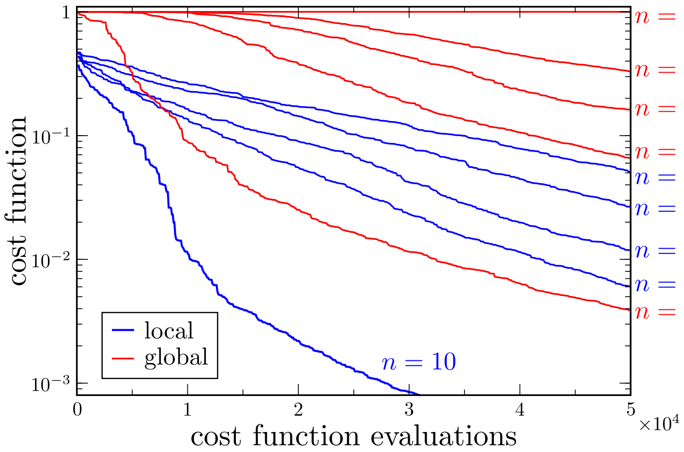

FIGURE 5.2: Comparison of local C_L and global C_G cost performance. Here we consider the QLSP of Eq. 5.28 for different system sizes. In all cases $\kappa = 20$. For each $n \in \{10, \dots, 50\}$, we plot the cost value versus the number of cost function evaluations. As n increases it becomes increasingly hard to train to global cost function. At $n = 50$, our optimization cannot significantly lower C_G below a value of one. On the other hand, we are able to train C_L for all values of n considered.

non-negative. Furthermore, it is clear that if $A|x\rangle = |b\rangle$, then we have $\widehat{C}_G = C_G = \widehat{C}_L = C_L = 0$. Conversely, assuming that $A|x\rangle \neq |b\rangle$ implies that $C_G > 0$ and hence that all four cost functions are positive. Therefore, all four cost functions are faithful, vanishing if and only if $A|x\rangle = |b\rangle$.

For illustrative purposes, in Figure 5.2 we show that, as n increases, it becomes increasingly hard to optimize the global cost function C_G. On the other hand, the local cost function C_L performs significantly better, as we are able to train C_L for systems of size up to $2^{50} \times 2^{50}$ (*i.e.*, with 50 qubits). These results show that the vanishing gradients of global cost functions could make them untrainable for large n, and hence we propose using our local cost functions for large-scale implementations.

5.1.2 *Operational meaning of cost functions*

Here we provide operational meanings for the aforementioned cost functions. These operational meanings are crucial since they allow one to define termination conditions for VQLS in order to achieve a desired precision. In particular, we find that the following bounds hold in general:

$$\widehat{C}_G \geq \frac{\epsilon^2}{\kappa^2}, \quad C_G \geq \frac{\epsilon^2}{\kappa^2}, \quad \widehat{C}_L \geq \frac{1}{n}\frac{\epsilon^2}{\kappa^2}, \quad C_L \geq \frac{1}{n}\frac{\epsilon^2}{\kappa^2}. \tag{5.10}$$

Note that one can take the right-hand-sides of these inequalities as the γ quantity shown in Figure 5.1.

We remark that, for C_G and C_L, the bounds in Eq. 5.10 can be tightened (by using the bounds on $\widehat{C}_G$ and $\widehat{C}_L$ in Eq. 5.10) as follows:

$$C_G \geq \frac{\epsilon^2}{\kappa^2 \langle\psi|\psi\rangle}, \quad C_L \geq \frac{1}{n}\frac{\epsilon^2}{\kappa^2\langle\psi|\psi\rangle}. \tag{5.11}$$

Here, $\langle\psi|\psi\rangle$ is experimentally computable (see Eq. 5.15 below) and satisfies $\langle\psi|\psi\rangle \leq 1$. Hence, when training C_G or C_L, one can employ the right-hand-sides of Eq. 5.11 as opposed to those of Eq. 5.10 as the termination condition γ.

[NC11] Nielsen and Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*

Furthermore, one can employ the operational meaning of the trace distance [NC11] to note that, for any POVM element M, we have $\epsilon \geq D(M)$, where

$$D(M) = |\langle x|M|x\rangle - \langle x_0|M|x_0\rangle| \tag{5.12}$$

measures the difference between expectation values on $|x\rangle$ and $|x_0\rangle$.

Let us now provide a proof for Eq. 5.10. Consider first that $\widehat{C}_G = \langle H_G\rangle$, with the eigenstates and eigenvalues of H_G denoted by $\{|x_i\rangle\}$ and $\{E_i\}$, respectively for $i = 0, 1, \ldots$. By construction $|x_0\rangle$ is the ground state of H_G with $E_0 = 0$. In what follows we assume for simplicity that $|x_1\rangle$ is non-degenerate, although the same proof approach works for the degenerate case.

It is clear that for a given ϵ, the smallest energy $\langle H_G\rangle$ (hence cost) is achieved if the state $|x\rangle$ is a superposition of $|x_0\rangle$ and $|x_1\rangle$ only. One can see this by expanding an arbitrary state $|x\rangle$ in the energy eigenbasis, $|x\rangle = \sum_i \chi_i |x_i\rangle$, and noting that ϵ depends only on the magnitude of χ_0. Hence for a fixed ϵ, one is free to vary the set of coefficients $\{\chi_i\}_{i\neq 0}$, and the set that minimizes the energy corresponds to choosing $\chi_i = 0$ for all $i > 1$.

So we take:

$$|x\rangle = \cos(\theta/2)|x_0\rangle + e^{i\phi}\sin(\theta/2)|x_1\rangle\,, \tag{5.13}$$

and the associated energy is given by

$$\langle x|H_G|x\rangle = E_1 \sin^2(\theta/2) \geq \frac{\sin^2(\theta/2)}{\kappa^2}\,, \tag{5.14}$$

[SSO19] Subaşı et al., "Quantum Algorithms for Systems of Linear Equations Inspired by Adiabatic Quantum Computing"

where we used the fact that $E_0 = 0$, and that the first excited state energy satisfies $E_1 \geq 1/\kappa^2$ (which was shown in Ref. [SSO19]). The trace distance between $|x\rangle$ and $|x_0\rangle$ can be easily computed as $\sqrt{1-|\langle x|x_0\rangle|^2}$, which results in $\epsilon = |\sin(\theta/2)|$. Inserting this into Eq. 5.14 yields $\widehat{C}_G \geq \epsilon^2/\kappa^2$. The remaining inequalities in Eq. 5.10 follow from Eq. 5.8 and from the fact that $\langle\psi|\psi\rangle \leq 1$, which implies $C_G \geq \widehat{C}_G$.

5.1.3 *Cost evaluation*

In principle, all the aforementioned cost functions can be efficiently evaluated using the Hadamard Test circuit and simple classical post-processing. However, in practice, care must be taken to minimize the number of controlled operations in these circuits. Consider evaluating the term $\langle\psi|\psi\rangle$, which can be written as

$$\langle\psi|\psi\rangle = \sum_{ll'} c_l c_{l'}^* \beta_{ll'}\,, \tag{5.15}$$

with

$$\beta_{ll'} = \langle\vec{0}|V^\dagger A_{l'}^\dagger A_l V|\vec{0}\rangle. \tag{5.16}$$

There are $L(L-1)/2$ different $\beta_{ll'}$ terms that one needs to estimate and which can be measured with Hadamard Tests. The Hadamard Test involves acting with V on $|\vec{0}\rangle$, and then using an ancilla as the control qubit, applying C_{A_l} followed by $C_{A_{l'}^\dagger}$, where C_W denotes controlled-W (see Figure 5.3 for precise circuits).

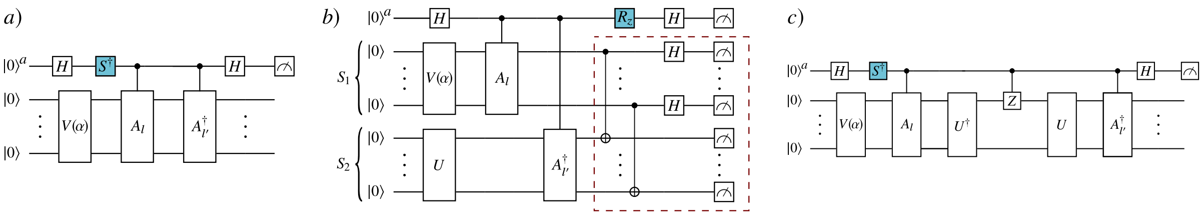

FIGURE 5.3: a) Circuit for the Hadamard Test used to compute the coefficients $\beta_{ll'} = \langle\vec{0}|V^\dagger A^\dagger_{l'} A_l V|\vec{0}\rangle$ and calculate the inner product $\langle\psi|\psi\rangle$ of 5.15. The phase gate in the colored box is excluded when calculating the real part of $\beta_{ll'}$ and included when calculating its imaginary part. b) Hadamard-Overlap Test used to compute the coefficients $\gamma_{ll'}$ defined in 5.18. The Overlap circuit of Refs. [GECP13; Cin+18] is indicated in the dashed box. Here, the R_z gate in the colored box denotes a rotation about the z axis of an angle $-\pi/2$. Excluding (including) this rotation allows one to calculate the real (imaginary) part of $\gamma_{ll'}$. As explained in the text, additional post-processing is required. c) Hadamard Test circuit for computing $\delta^{(j)}_{ll'}$ as defined in 5.21. Shown here is case when $j = 1$.

In addition, for $\widehat{C}_G$ and C_G, one needs to evaluate

$$|\langle b|\psi\rangle|^2 = |\langle\vec{0}|U^\dagger AV|\vec{0}\rangle|^2 = \sum_{ll'} c_l c^*_{l'} \gamma_{ll'}\,, \tag{5.17}$$

with

$$\gamma_{ll'} = \langle\vec{0}|U^\dagger A_l V|\vec{0}\rangle\langle\vec{0}|V^\dagger A^\dagger_{l'} U|\vec{0}\rangle. \tag{5.18}$$

The γ_{ll} terms are easily estimated by applying $U^\dagger A_l V$ to $|\vec{0}\rangle$ and then measuring the probability of the all-zeros outcome. For the $L(L-1)/2$ terms with $l \neq l'$, there are various strategies to estimate $\gamma_{ll'}$. For example, one could estimate the L terms of the form $\langle\vec{0}|U^\dagger A_l V|\vec{0}\rangle$ with a Hadamard Test, but one would have to control all of the unitaries: V, A_l, and $U^\dagger$. Instead, we introduce a novel circuit called the Hadamard-Overlap Test that directly computes $\gamma_{ll'}$ without having to control V or U at the expense of doubling the number of qubits. This circuit is shown in Figure 5.3.

Finally, for $\widehat{C}_L$ and C_L, one needs to estimate terms of the form

$$\delta^{(j)}_{ll'} = \langle\vec{0}|V^\dagger A^\dagger_{l'} U(|0_j\rangle\langle 0_j| \otimes \mathbb{1}_{\bar{j}})U^\dagger A_l V|\vec{0}\rangle\,. \tag{5.19}$$

These terms can either be estimated with the Hadamard-Overlap Test or with the Hadamard Test, which are discussed below.

▸ HADAMARD TEST. Figure 5.3(a) shows a Hadamard Test which can be used to measure the coefficients $\beta_{ll'}$ defined in 5.16, and used to compute Eq. $\langle\psi|\psi\rangle$ as in Eq. 5.15. When the phase gate is excluded, the probability of measuring the ancilla qubit in the $|0\rangle^a$ state is $P(0) = (1 + \mathrm{Re}[\beta_{ll'}])/2$, while the probability of measuring it in the $|1\rangle^a$ state is $P(1) = (1 - \mathrm{Re}[\beta_{ll'}])/2$. Hence, by means of the Hadamard Test we can compute the real part of $\beta_{ll'}$ as

$$\mathrm{Re}[\beta_{ll'}] = P(0) - P(1)\,. \tag{5.20}$$

With a similar argument, it can be easily shown that by including the phase gate, one can compute $\mathrm{Im}[\beta_{ll'}]$.

As we now show, in order to compute the coefficients $\delta^{(j)}_{ll'}$ in Eq. 5.19 we can use the previous result combined with those obtained by means of the Hadamard test of Figure 5.3(c). In particular, since $|0_j\rangle\langle 0_j| = (\mathbb{1}_j + Z_j)/2$, then we can express

$$\delta^{(j)}_{ll'} = \beta_{ll'} + \langle\vec{0}|V^\dagger A^\dagger_{l'} U(Z_j \otimes \mathbb{1}_{\bar{j}})U^\dagger A_l V|\vec{0}\rangle\,. \tag{5.21}$$

Hence, in order to calculate $\delta_{ll'}^{(j)}$ one only needs to measure the real and imaginary parts of the matrix elements $\langle\vec{0}|V^\dagger A_{l'}^\dagger U(Z_j\otimes\mathbb{1}_{\bar{j}})U^\dagger A_l V|\vec{0}\rangle$, which can be accomplished by means of the circuit in Figure 5.3(c).

- HADAMARD-OVERLAP TEST. Consider the circuit in Figure 5.3(b), which we refer to as the Hadamard-Overlap Test. A nice feature of the Hadamard-Overlap Test is that it only requires one application of both U and V, and these unitaries do not need to be controlled, in contrast to the Hadamard Test. As explained below, the circuit for the Hadamard-Overlap Test can be obtained by combining the Hadamard Test with the Overlap circuit of Refs. [GECP13; Cin+18]. This circuit requires $2n+1$ qubits and classical post-processing (which scales linearly with n) similar to that of the Overlap circuit, except that here we add a conditional statement.

[GECP13] Garcia-Escartin and Chamorro-Posada, "Swap test and Hong-Ou-Mandel effect are equivalent"

[Cin+18] Cincio et al., "Learning the quantum algorithm for state overlap"

When the R_z gate in Figure 5.3(b) is excluded, and conditioning the measurement on the ancilla qubit to yield the $|0\rangle^a$ state, we can perform the depth-two Overlap circuit between registers S_1 and S_2 to get

$$\begin{aligned}P(0)=\frac{1}{2}(&\langle\vec{0}|U^\dagger V|\vec{0}\rangle\langle\vec{0}|V^\dagger U|\vec{0}\rangle\\&+\langle\vec{0}|U^\dagger A_{l'}A_l V|\vec{0}\rangle\langle\vec{0}|V^\dagger A_l^\dagger A_{l'}^\dagger U|\vec{0}\rangle\\&+\mathrm{Re}\left[\langle\vec{0}|U^\dagger A_l V|\vec{0}\rangle\langle\vec{0}|V^\dagger A_{l'}^\dagger U|\vec{0}\rangle\right]).\end{aligned}\tag{5.22}$$

On the other hand, by conditioning the ancilla qubit being measured in the state $|1\rangle^a$, we perform the Overlap circuit between subsystems S_1 and S_2 to obtain

$$\begin{aligned}P(1)=\frac{1}{2}(&\langle\vec{0}|U^\dagger V|\vec{0}\rangle\langle\vec{0}|V^\dagger U|\vec{0}\rangle\\&+\langle\vec{0}|U^\dagger A_{l'}A_l V|\vec{0}\rangle\langle\vec{0}|V^\dagger A_l^\dagger A_{l'}^\dagger U|\vec{0}\rangle\\&-\mathrm{Re}\left[\langle\vec{0}|U^\dagger A_l V|\vec{0}\rangle\langle\vec{0}|V^\dagger A_{l'}^\dagger U|\vec{0}\rangle\right]).\end{aligned}\tag{5.23}$$

Then, combining Eqs. 5.22 and 5.23 yields

$$\mathrm{Re}[\gamma_{ll'}]=P(0)-P(1).\tag{5.24}$$

Following a similar procedure, it can be shown that including the R_z gate allows us to calculate $\mathrm{Im}[\gamma_{ll'}]$.

Note that the Hadamard-Overlap test can also be used to compute the real and imaginary parts of $\delta_{ll'}^{(j)}$ in Eq. 5.19. In this case an additional random unitary R_j must be initially applied to the qubits in register S_2 in order to generate the input state $|0_j\rangle\langle 0_j|\otimes\mathbb{1}_{\bar{j}}$. Specifically, R_j randomly applies a bit-flips to all qubits except qubit j:

$$R_j=X_1^{r_1}\otimes X_2^{r_2}\otimes\cdots\otimes\mathbb{1}_j^{r_j}\otimes\cdots\otimes X_n^{r_n},\tag{5.25}$$

with $\vec{r}=r_1,r_2\ldots,r_n$ a random bitstring of length n.

5.1.4 *Ansatz*

In the VQLS algorithm, $|x\rangle$ is prepared by acting on the $|\vec{0}\rangle$ state with a trainable gate sequence $V(\vec{\alpha})$. Without loss of generality, $V(\vec{\alpha})$ can be expressed in terms of L gates from a gate alphabet $\mathcal{A}=\{G_k(\alpha)\}$ as

$$V(\vec{\alpha})=G_{k_L}(\alpha_L)\ldots G_{k_i}(\alpha_i)\ldots G_{k_1}(\alpha_1).\tag{5.26}$$

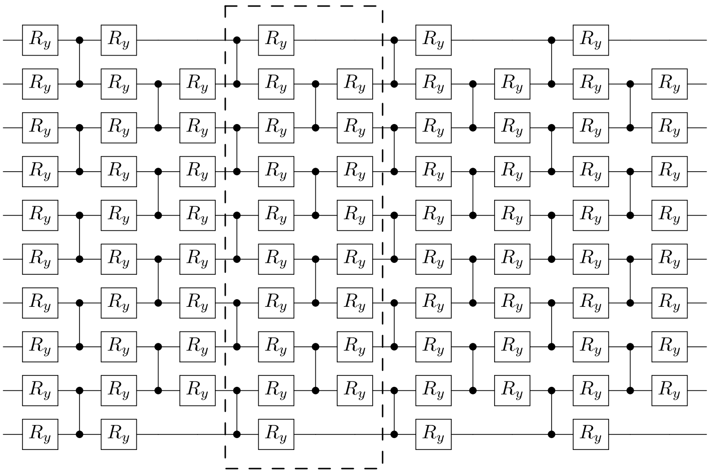

FIGURE 5.4: Fixed-structure layered Hardware-Eficient Ansatz for $V(\vec{\alpha})$. As indicated by the dashed box, each layer is composed of controlled-Z gates acting on alternating pairs of neighboring qubits which are preceded and followed by single qubit rotations around the y-axis, $R_y(\alpha_i) = e^{-i\alpha_i Y/2}$. Shown is the case of four layers and $n = 10$ qubits. The number of variational parameters and gates scales linearly with n: for $n = 50$, four layers of this ansatz consist of 640 gates and 440 variational parameters.

Here $\vec{k} = (k_L, \dots, k_1)$ identifies the types of gates and their placement in the circuit (*i.e.*, on which qubit they act), while $\vec{\alpha}$ are continuous parameters. When working with specific quantum hardware, it is convenient to choose a Hardware-Efficient Ansatz [Kan+17], where $\mathcal{A}$ is composed of gates native to that hardware. This reduces the gate overhead that arises when implementing the algorithm in the actual device. We use the term “fixed-structure ansatz” when the gate structure of $V(\vec{\alpha})$ is fixed (*i.e.*, when $\vec{k}$ is fixed), and when one only optimizes over $\vec{\alpha}$. Figure 5.4 shows an example of such an ansatz, with $\mathcal{A}$ composed of single-qubit y-rotations and controlled-Z gates. We employ the ansatz in Figure 5.4 for the heuristics in Section 5.2.1. Let us remark that this ansatz can have trainability issues [McC+18; Cer+21b] for large-scale problems.

[Kan+17] Kandala et al., “Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets”

[McC+18] McClean et al., “Barren plateaus in quantum neural network training landscapes”

[Cer+21b] Cerezo et al., “Cost function dependent barren plateaus in shallow parametrized quantum circuits”

Strategies such as layer-by-layer training [Gra+19] and correlating the $\vec{\alpha}$ parameters [VC21] have been shown to be effective in addressing these trainability issues. In addition, trainability could be further improved by combining these strategies with more advanced ansatz architectures, and we now consider two such architectures. First, we discuss a “variable structure ansatz” [Cin+18; LaR+18], where one optimizes over the gate angles and the gate placement in the circuit, *i.e.*, where one optimizes over $\vec{\alpha}$ and also over $\vec{k}$. We employ such ansatz for our heuristics in Section 5.2.2. We refer to the section below for a discussion of the optimization method employed for a variable structure ansatz.

[Gra+19] Grant et al., “An initialization strategy for addressing barren plateaus in parametrized quantum circuits”

[VC21] Volkoff and Coles, “Large gradients via correlation in random parameterized quantum circuits”

[Cin+18] Cincio et al., “Learning the quantum algorithm for state overlap”

[LaR+18] LaRose et al., “Variational quantum state diagonalization”

In addition to the aforementioned ansatz, one can also employ the Quantum Alternating Operator Ansatz (QAOA) [FGG14; Had+19] to construct the unitary $V(\vec{\alpha})$. As we discussed in Chapter 1, the QAOA consists of evolving the $H^{\otimes n}|\vec{0}\rangle$ state (where H denotes the Hadamard unitary) by two Hamiltonians for a specified number of layers, or rounds. These Hamiltonians are conventionally known as *driver* and *mixer* Hamiltonians, and respectively denoted as H_D and H_M. Since the ground state of both H_G and H_L is $|x_0\rangle$, we can either use 5.4 or 5.7 as the driver Hamiltonian H_D. Evolving with H_D for

[FGG14] Farhi et al., “A quantum approximate optimization algorithm”

[Had+19] Hadfield et al., “From the quantum approximate optimization algorithm to a quantum alternating operator ansatz” a time α_i corresponds to the unitary operator $U_D(\alpha_i) := e^{-iH_D\alpha_i}$. Moreover, one can take the mixer Hamiltonian to be the conventional $H_M = \prod_{i=1}^{n} X_i$, where X_i denotes Pauli X acting on the ith qubit. Accordingly, evolving with H_M for a time α_j yields the unitary operator $U_M(\alpha_j) := e^{-iH_M\alpha_j}$. The trainable ansatz $V(\vec{\alpha})$ is then obtained by alternating the unitary operators $U_D(\alpha_i)$ and $U_M(\alpha_j)$ p times:

$$V(\vec{\alpha}) = e^{-iH_M\alpha_{2p}} e^{-iH_D\alpha_{2p-1}} \cdots e^{-iH_M\alpha_2} e^{-iH_D\alpha_1}. \tag{5.27}$$

In this ansatz, each α_i is a trainable continuous parameter. We note that QAOA is known to be universal as the number of layers p tends to infinity [FGG14; Llo18], and that finite values of p have obtained good results for several problems [Wan+18; Cro18; Zho+20b]. In the Appendix C we present results of a small-scale implementation of VQLS with a QAOA ansatz.

[FGG14] Farhi et al., "A quantum approximate optimization algorithm"

[Llo18] Lloyd, "Quantum approximate optimization is computationally universal"

[Wan+18] Wang et al., "Quantum approximate optimization algorithm for MaxCut: A fermionic view"

[Cro18] Crooks, "Performance of the quantum approximate optimization algorithm on the maximum cut problem"

[Zho+20b] Zhou et al., "Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices"

Let us remark that Ref. [HHL09] showed that it is possible to efficiently generate an accurate approximation to the true solution $|x_0\rangle$, *i.e.*, with a number of gates that is polynomial in n, assuming certain constraints on A and $\vec{b}$. Therefore, in principle, one may efficiently approximate these solutions with a universal variational ansatz, such as the ones discussed above.

[HHL09] Harrow et al., "Quantum algorithm for linear systems of equations"

5.1.5 *Training algorithm*

Several classical optimizers may be employed to train $V(\vec{\alpha})$ and minimize the cost functions of VQLS. For example, our heuristics in Section 5.2.1 employ an optimization method that, at each iteration, chooses a random direction $\vec{w}$ in the parameter space along which to perform a line search, *i.e.*, to solve $\min_{s\in\mathbb{R}} C(\vec{\alpha}+s\vec{w})$. On the other hand, in Section 5.2.2 we perform an optimization where all the parameters in $\vec{\alpha}$ are independently optimized at each iteration.

Let us discuss in more detail the optimization method employed for the heuristics in Section 5.2.2. As mentioned previously, we employed a variable-structure ansatz where the gate placement and the type of gates in $V(\vec{\alpha})$ can change during the optimization. Our approach here is similar to the variable-structure ansatzes employed in Refs. [Cin+18; LaR+18].

[Cin+18] Cincio et al., "Learning the quantum algorithm for state overlap"

[LaR+18] LaRose et al., "Variational quantum state diagonalization"

First, the gate structure and the angles of $V(\vec{\alpha})$ are randomly initialized. That is, one randomly chooses $\vec{k}$, and $\vec{\alpha}$ in Eq. 5.26. Then, the optimization is performed in two alternating loops: an inner loop and an outer loop. During the inner loop, $\vec{k}$ is fixed, and one optimizes over $\vec{\alpha}$. Once a local minimum is reached, the outer optimization loop changes the circuit layout. In this outer loop, the circuit is randomly grown by inserting into $V(\vec{\alpha})$ a sequence of parametrized gates which compile to identity, such that they do not change the cost value. The previous process is then repeated by alternating between the inner and outer loops until the optimization termination condition is met.

Here we remark that the goal of the outer loop is to enhance the expressivity of $V(\vec{\alpha})$ and lead to smaller cost values during the next inner loop. However, it may happen that after growing the circuit, the optimizer is not able to minimize the cost function. This is due to the fact that some gate insertions do not lead to more expressive circuits. In order to avoid such unnecessary circuit growth, one can then accept the parametrized gate insertion conditioned to lead to smaller cost values.

5.2 HEURISTIC SCALING

Here we study the scaling of VQLS with the condition number κ, error tolerance ϵ, and number of qubits n. First, we consider a specific QLSP for which $|x_0\rangle$ admits an efficient matrix-product-state representation, allowing us to simulate large values of n. We then consider QLSPs where the matrix A is randomly generated. In both cases, we restrict A to be a sparse matrix, which is standard for QLSPs [HHL09], and we simulate VQLS without finite sampling. Moreover, we quantify the run time of VQLS with the *time-to-solution*, which refers to the number of exact cost function evaluations during the optimization needed to guarantee that ϵ is below a specified value. In practice, for large-scale implementations where the true solution $|x_0\rangle$ is unknown, ϵ cannot be directly calculated. Instead, one can use the operational meaning of our cost function in Eq. 5.10 to upper-bound ϵ. Hence, we take this approach in all of our heuristics, *i.e.*, we use the value of the cost, combined with Eq. 5.10 to determine the worst-case ϵ. We emphasize that, while it is tempting to directly compute ϵ from Eq. 5.2 in one's heuristics, this is essentially cheating since $|x_0\rangle$ is unknown, and this is why our certification procedure is so important.

[HHL09] Harrow et al., "Quantum algorithm for linear systems of equations"

5.2.1 *Ising-inspired quantum linear system problem*

Here we numerically simulate VQLS to solve the QLSP defined by the sparse matrix

$$\begin{aligned} A &= \frac{1}{\zeta}\left(\sum_{j=1}^{n} X_j + J\sum_{j=1}^{n-1} Z_j Z_{j+1} + \eta \mathbb{1}\right), \\ |b\rangle &= H^{\otimes n}|\vec{0}\rangle , \end{aligned} \tag{5.28}$$

where the subscripts in 5.28 denote the qubits acted upon non-trivially by the Pauli operator. Here, we set $J = 0.1$. The parameters ζ and η are chosen such that the smallest eigenvalue of A is $1/\kappa$ and its largest eigenvalue is 1, which involves analytically computing [HG17] the smallest eigenvalue of the first two terms of A and then re-scaling A. As previously mentioned, this QLSP example is motivated from the fact that for $J = 0$ the solution is given by $|x_0\rangle = |b\rangle$. Hence for small J, $|x_0\rangle$ admits an efficient matrix-product-state representation.

[HG17] He and Guo, "The boundary effects of transverse field Ising model"

- Dependence on κ. Figure 5.5 shows our results, plotting time-to-solution versus κ for the QLSP in 5.28. Our numerical results were obtained by employing the layered Hardware-Efficient Ansatz of Figure 5.4, and by training the local cost C_L for different values of n. Figure 5.5 shows that as the condition number κ is increased, the time-to-solution needed to achieve a given ϵ increases with a scaling that appears to be sub-linear. Hence VQLS scales efficiently with κ for this example. It is known that linear scaling is optimal [HHL09]. Hence we expect that the scaling observed here is specific to this example, and indeed the example in the next subsection shows scaling that is closer to linear.

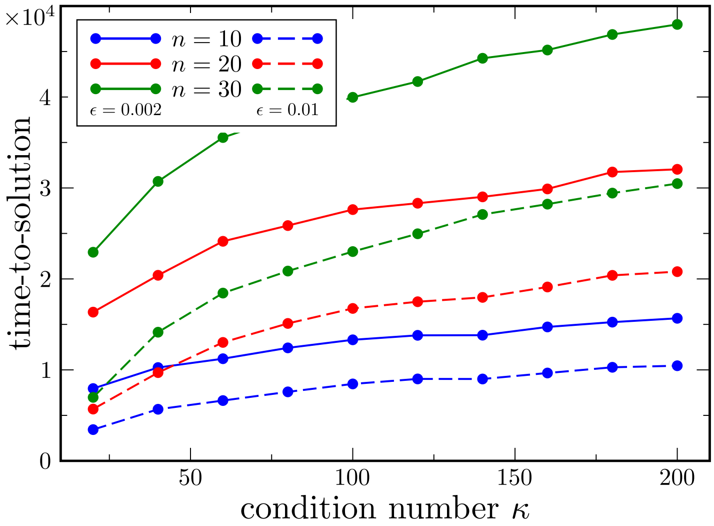

FIGURE 5.5: Scaling with κ for the Ising-inspired QLSP. The time-to-solution is the number of executions needed to guarantee a precision of $\epsilon = 0.002$ (solid line) and $\epsilon = 0.01$ (dashed line). Curves are shown for $n = 10, 20, 30$ qubits. In each case, we averaged over 30 runs of the VQLS algorithm with four layers of the Layered Hardware-Efficient Ansatz of Figure 5.4, and we trained the gate sequence by minimizing C_L of Eq. 5.6. While the κ scaling appears to be sub-linear here, it is known that linear scaling is optimal in general, and hence the observed scaling is likely specific to this example.

- **DEPENDENCE ON ϵ.** To study the scaling of VQLS with ϵ, we numerically solved the QLSP in 5.28 for different values of κ and n. In all cases, we trained the gate parameters by optimizing the C_L cost function. Figure 5.6 shows the time-to-solution versus $1/\epsilon$. These results show that as $1/\epsilon$ grows, the time-to-solution exhibits a logarithmic growth.

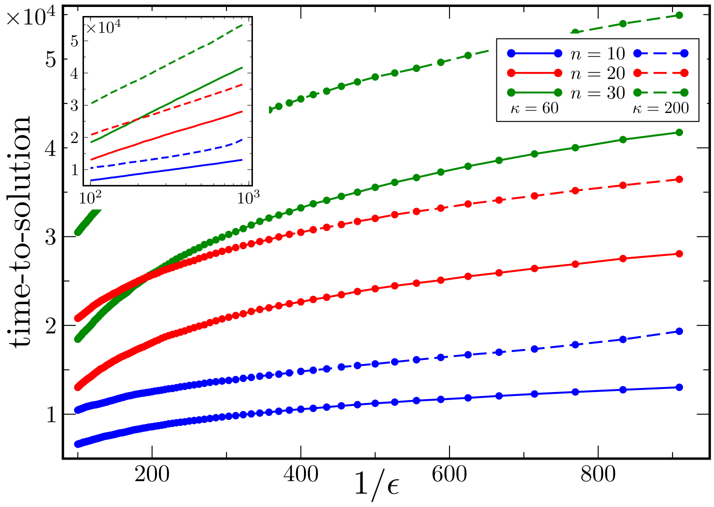

FIGURE 5.6: Scaling with $1/\epsilon$ for the Ising-inspired QLSP. Curves are shown for $n = 10, 20, 30$ qubits, with $\kappa = 60$ (solid line) or $\kappa = 200$ (dashed line). In all cases $V(\vec{\alpha})$ was composed of four layers of the Layered Hardware-Efficient Ansatz of Figure 5.4, and we trained the local cost C_L. The time-to-solution was obtained by averaging over 30 runs of the VQLS algorithm. The inset depicts the same data on a logarithmic scale. The dependence on $1/\epsilon$ appears to be logarithmic, *i.e.*, linear on a logarithmic scale.

- DEPENDENCE ON n. The QLSP of 5.28 allows us to increase the number of qubits and analyze the scaling of VQLS with n. Here we implemented VQLS with $n = 6, 8, \ldots, 30$ and for $\kappa = 60, 120, 200$ by training the local cost function C_L. Figure 5.7 shows time-to-solution versus n. As the number of qubits increases, the time-to-solution needed to guarantee a particular ϵ with κ fixed appears to increase linearly with n. This corresponds to logarithmic scaling in the linear system size N, analogous to that of the HHL algorithm [HHL09].

[HHL09] Harrow et al., "Quantum algorithm for linear systems of equations"

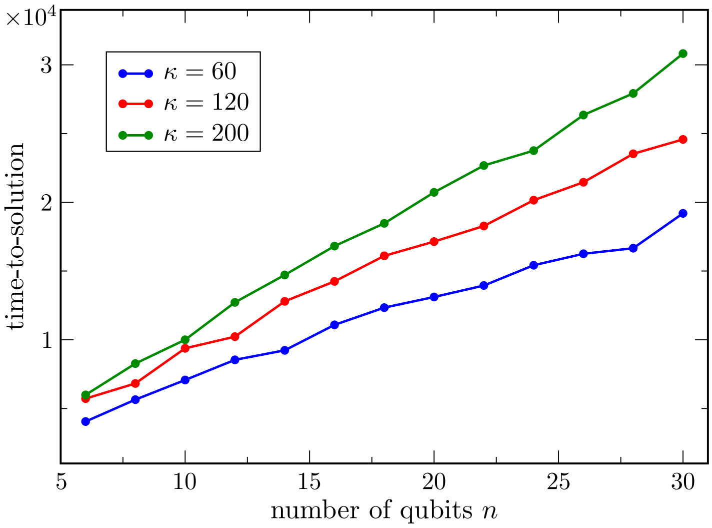

FIGURE 5.7: Scaling with n for the Ising-inspired QLSP. Curves are shown for $\epsilon = 0.01$ and for $\kappa = 60, 120, 200$. In all cases we trained the local cost C_L with four layers of the Layered Hardware-Efficient Ansatz of Figure 5.4. The dependence on n appears to be linear (logarithmic in N) for this example.

5.2.2 *Randomly-generated quantum linear system problem*

In this section, we present scaling results for the case when the matrix A is randomly generated with the form

$$A = \xi_1 \Big(\mathbb{1} + \xi_2 \sum_j \sum_{k \neq j} p a_{j,k} \sigma_j^\alpha \sigma_k^\beta \Big). \tag{5.29}$$

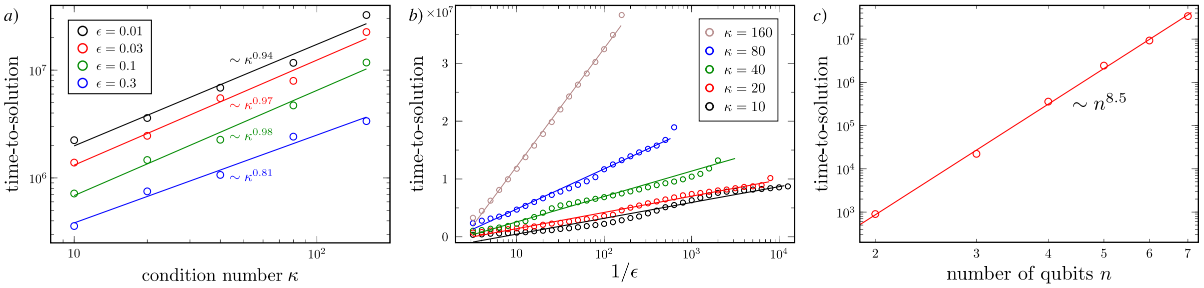

FIGURE 5.8: VQLS heuristic scaling for random matrices generated according to 5.29. The time-to-solution is the number of executions needed to guarantee a desired precision ϵ. In all cases we employed a variable-structure ansatz $V(\vec{\alpha})$, and we trained the local cost C_L of Eq. 5.6. a) Time-to-solution versus κ for a system of $n = 4$ qubits. Axes are shown on a log-log scale. For each value of ϵ the data were fitted with a power function κ^m and in all cases $m < 1$, suggesting that the κ scaling appears to be sub-linear. b) Time-to-solution versus $1/\epsilon$ for a system of $n = 4$ qubits. The x axis is shown in a log scale. Each curve corresponds to a different condition number. For all values of κ, the data were fitted with a linear function, implying that the $1/\epsilon$ scaling is logarithmic. c) Time-to-solution versus n needed to guarantee $\epsilon = 0.3$. All matrices had a condition number $\kappa = 10$. The plot employs a log-log scale. The data were fitted with a power function $y \sim n^{8.5}$, suggesting that the N dependence is polylogarithmic.

Here p is either 0 or 1 according to a fixed binomial distribution, $a_{j,k}$ are random weights in $(-1,1)$, and σ_j^α is the Pauli matrix acting on qubit j with $\alpha = x, y, z$. For each $j, k = 1, \ldots, n$ in 5.29, α and β are randomly chosen. Finally, we remark that ξ_1, and ξ_2 are normalization coefficients that rescale the matrix so that its largest eigenvalue is 1 and its smallest is $1/\kappa$ (where κ is fixed).

For a given number of qubits n, we randomly created a matrix A according to 5.29, and we ran four independent instances of VQLS. We then selected the best run, *i.e.*, the instance that required the smallest number of cost function evaluations to reach a specified value of guaranteed ϵ (guaranteed via Eq. 5.10). This procedure was then repeated for 10 independent random matrices A, and the time-to-solution was obtained as the average of the best run for each matrix.

- DEPENDENCE ON κ. In Figure 5.8(a) we show the time-to-solution versus κ for matrices randomly generated according to 5.29, and for $n = 4$. Here we employed a variable-structure ansatz as described in Section 5.1.5, and we trained the local cost in Eq. 5.6. Different curves represent different desired precision ϵ. The data were plotted in a log-log scale, and each curve was fitted with a power function κ^m. In all cases, we found $m < 1$, indicating that the scaling in κ for these examples is at worst linear. Linear scaling in κ is known to be optimal [HHL09].

[HHL09] Harrow et al., "Quantum algorithm for linear systems of equations"

- DEPENDENCE ON ϵ. Let us now analyze the scaling of VQLS with respect to ϵ for matrices with different condition numbers. Figure 5.8(b) depicts the time-to-solution versus ϵ for matrices randomly generated according to 5.29, and for $n = 4$. All curves were fitted with a linear function, and since the x axis is in a logarithmic scale, the dependence on $1/\epsilon$ appears to be logarithmic.

- DEPENDENCE ON n. In Figure 5.8(c) we present the time-to-solution versus n needed to guarantee $\epsilon = 0.3$ for QLSPs with $n = 2, \ldots, 7$. All matrices A had condition number $\kappa = 10$. The data were fitted with a power function, and we obtained the relation $y \sim n^{8.5}$. This corresponds to polylogarithmic scaling in N, which is the standard goal of quantum algorithms for the QLSP [Amb10; CKS17; CGJ18; SSO19].

We refer to the Appendix C for additional numerical simulations of VQLS for other QLSP examples, both with a Hardware-Efficient Ansatz and with a QAOA ansatz. These examples also exhibit efficient scaling behavior.

[Amb10] Ambainis, "Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations"

[CKS17] Childs et al., "Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision"

[CGJ18] Chakraborty et al., "The power of block-encoded matrix powers: improved regression techniques via faster Hamiltonian simulation"

[SSO19] Subaşı et al., "Quantum Algorithms for Systems of Linear Equations Inspired by Adiabatic Quantum Computing"

5.3 IMPLEMENTATION ON QUANTUM HARDWARE

Here we present the results of a 1024 × 1024 (*i.e.*, 10-qubit) implementation of VQLS using Rigetti's *16Q Aspen-4* quantum computer. Specifically, we solved the QLSP defined by the matrix A in 5.28, with $\zeta = \eta = 1$, and where the vector $|b\rangle = |0\rangle$ was the all-zero state. The ansatz consisted of $R_y(\alpha_i)$ gates acting on each qubit. To adapt to hardware constraints, we computed the cost function C_G in Eq. 5.5 by expanding the effective Hamiltonian H_G in terms of Pauli operators and then employing Rigetti's quantum computer to estimate the expectation values of these terms.

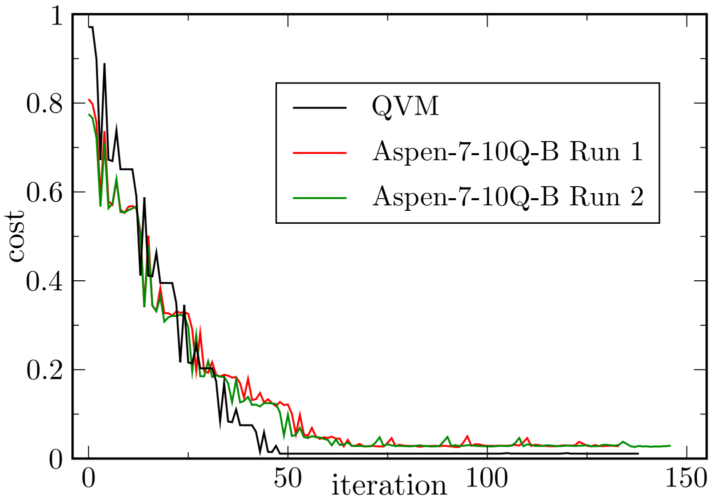

FIGURE 5.9: Implementation of VQLS on Rigetti's quantum hardware. Cost function C_G is plotted against the number of optimization steps, where A is defined in 5.28. One can observe that for each QPU run the cost function is reduced to a value below 10^{-1}. Due to noise present in the quantum device the cost does not go to zero.

The results of two representative VQLS runs are shown in Figure 5.9. As shown, the cost function data obtained by training in a quantum computer closely matches the one obtained from training on a noiseless simulator. For each run on the QPU, the cost function value approaches zero, indicating that a good solution to the linear system was found.

Additional experiments performed on quantum hardware are presented in the Appendix C.

5.4 OUTLOOK

This chapter presented a variational quantum algorithm called VQLS for solving the quantum linear systems problem. On the analytical side, we presented four different faithful cost functions, we derived efficient quantum circuits to estimate them while showing that they are difficult to estimate classically, and we proved operational meanings for them as upper bounds on ϵ^2/κ^2. On the numerical side, we studied the scaling of the VQLS run time by solving non-trivial problems of size up to $2^{50} \times 2^{50}$. We found VQLS to scale efficiently for the examples considered, namely, at worst linearly in κ, logarithmically in $1/\epsilon$, and polylogarithmically in the linear system size N.

It remains to be seen how the VQLS training is affected by finite sampling, which is not accounted for in our heuristics. Our solution verification procedure in Section 5.1.2 will require the shot noise to appropriately scale with ϵ and κ as dictated by Eq. 5.10. Namely, the number of shots would need to scale as $(\kappa/\epsilon)^4$, although this complexity might be reduced if one does not require solution verification.

Furthermore, we utilized Rigetti's Quantum Cloud Services to implement VQLS for a particular problem up to a size of 1024 × 1024, which to our knowledge is the largest implementation of a linear system on quantum hardware. Interestingly, with our implementation on Rigetti's hardware, we noticed some preliminary evidence of noise resilience, along the same lines as those discussed in Ref. [Sha+20b] for a different variational algorithm. Namely, we noticed optimal parameter resilience, where VQLS learned the correct optimal parameters despite various noise sources (*e.g.*, measurement noise, decoherence, gate infidelity) acting during the cost evaluation circuit.

Finally, we discuss how VQLS fits into the larger literature on quantum algorithms for linear systems. Most prior algorithms rely on time evolutions with the matrix A [HHL09; Amb10; CKS17] or a simple function of it [SSO19]. In these algorithms, the duration of the time evolution is $O(\kappa)$ in order to prepare a state $|x\rangle$ that is ϵ-close to the correct answer. In general, this can only be achieved with a quantum circuit of size linear in κ as per the "no fast-forwarding theorem" [Ber+07; AA17]. This is even true if there exists a very short quantum circuit that prepares the desired state $|x\rangle$. The non-variational algorithms simply cannot exploit this fact. On the other hand, a variational algorithm with a short-depth ansatz might be used to prepare such a state.

This does not mean, however, that the overall complexity of the variational algorithm does not depend on the condition number. This dependence enters through the stopping criteria given in Eq. 5.10. As the condition number increases, the cost has to be lowered further in order to guarantee an error of ϵ. This will undoubtedly require more iterations of the variational loop to achieve. In effect, our variational approach trades the gate complexity of non-variational algorithms with the number of iterations for a fixed circuit depth. This trade-off can be useful in utilizing NISQ devices without error correction.

We remark that other variational approaches to the QLSP distinct from this one were very recently proposed [Xu+21; HBR21]. Relatively speaking, the distinct aspects of this work include: (1) our quantitative certification procedure for the solution, (2) our clear approach to improve trainability for large-scale problems, (3) our novel circuits for efficient cost evaluation, (4) our large-scale heuristics demonstrating efficient scaling, and (5) our large-scale implementations on quantum hardware. Finally, it is exciting that after the realization of this work, Refs. [PJ+21a; PJ+21b; PWK22] studied the VQLS performance with different ansatz architectures and classical optimizers. Moreover, two independent tutorials for the VQLS algorithm were created and added to IBM's open-source Qiskit textbook [Asf+19], and to Xanadu's PennyLane library [Mar19].

[Sha+20b] Sharma et al., "Noise resilience of variational quantum compiling"

[HHL09] Harrow et al., "Quantum algorithm for linear systems of equations"

[Amb10] Ambainis, "Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations"

[CKS17] Childs et al., "Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision"

[SSO19] Subaşı et al., "Quantum Algorithms for Systems of Linear Equations Inspired by Adiabatic Quantum Computing"

[Ber+07] Berry et al., "Efficient quantum algorithms for simulating sparse Hamiltonians"

[AA17] Atia and Aharonov, "Fast-forwarding of Hamiltonians and exponentially precise measurements"

[Xu+21] Xu et al., "Variational algorithms for linear algebra"

[HBR21] Huang et al., "Near-term quantum algorithms for linear systems of equations with regression loss functions"

[PJ+21a] Pellow-Jarman et al., "A comparison of various classical optimizers for a variational quantum linear solver"

[PJ+21b] Pellow-Jarman et al., "Near Term Algorithms for Linear Systems of Equations"

[PWK22] Patil et al., "Variational quantum linear solver with a dynamic ansatz"

Part III

QUANTUM GENERATIVE MODELING

6

Quantum generative adversarial networks for Monte Carlo events

> *"It was the great multiplicity of the hadrons that led to the formulation of the quark model. Without some organizing principle such a large collection of particles seemed unwieldy, and the possibility that they might all be elementary offended those who hold the conviction, or at least the fond wish, that nature should be simple."*
> —Yoichiro Nambu

The current availability of noisy intermediate-scale quantum (NISQ) computers [Pre18], and recent advances towards quantum computational supremacy [Aru+19; Zho+20a], have led to a growing interest in these devices to perform computational tasks faster than classical machines. Among many of the near-term applications [Cer+21a; Bha+22], the field of Quantum Machine Learning (QML) [Bia+17; SP18] is held as one promising approach to make use of NISQ computers.

Early work in QML was mostly focused on speeding up linear algebra subroutines [WBL12; LMR13; RML14; KP20], widely used in classical machine learning, by leveraging the Harrow-Hassidim-Lloyd algorithm [HHL09]. This approach is promising, though its utility depends on the existence of large-scale quantum computers with low gate errors and enough qubits to perform quantum error correction. As discussed in previous chapters, more recent proposals focus on defining a quantum neural network (QNN), or parameterized quantum circuit [Ben+19c; SJAG19; BP+20; Lar+21], which then can be trained to implement a function class [SSM21; GTN21; PS+21a]; these proposals can be implemented on current NISQ-era devices. For example, several QNNs have been proposed for pattern classification [Hav+19; Sch+20; PS+20a; Dut+21] or data compression [ROA17; BP21; CW21; DT21]. This QML approach to quantum computing is a research topic that can be adapted, improved, and tested on many research problems in disparate scientific fields. Motivated by this idea, we propose to investigate the possibility of using QNNs for generative modeling [Ben+19a; HDP19; Coy+20; Coy+21]. More specifically, we explore the uses of QNNs for the generation of Monte Carlo events through quantum generative adversarial networks (qGANs) [DDK18; LW18].

The generative adversarial framework employs two competing networks, the generator and the discriminator, that are trained alternatively [Goo+14]. The generator produces candidates while the discriminator evaluates them. The objective of the discriminator is to distinguish the real samples from the generated ones. That is, the discriminator plays the role of the generator's adversary, and therefore, their competition is a zero-sum two-player game. By substituting either the discriminator, the generator, or both with quantum

[Aru+19] Arute et al., "Quantum supremacy using a programmable superconducting processor"

[Zho+20a] Zhong et al., "Quantum computational advantage using photons"

[Bia+17] Biamonte et al., "Quantum machine learning"

[SP18] Schuld and Petruccione, *Supervised learning with quantum computers*

[WBL12] Wiebe et al., "Quantum algorithm for data fitting"

[LMR13] Lloyd et al., "Quantum algorithms for supervised and unsupervised machine learning"

[RML14] Rebentrost et al., "Quantum Support Vector Machine for Big Data Classification"

[KP20] Kerenidis and Prakash, "Quantum gradient descent for linear systems and least squares"

[HHL09] Harrow et al., "Quantum algorithm for linear systems of equations"

[DDK18] Dallaire-Demers and Killoran, "Quantum generative adversarial networks"

[LW18] Lloyd and Weedbrook, "Quantum generative adversarial learning"

[Goo+14] Goodfellow et al., "Generative adversarial nets" systems, we translate the scheme to quantum computing.

Recently, the spreading interest in QML has led to different qGAN implementations [ZLW19; Zen+19; Hu+19; Ben+19b; Sit+20; RAG21; Niu+21]. The contribution here can be summarized in three distinct aspects. (1) Previous proposals employed toy data for their qGAN training. In contrast, we test our model using data for a quantum scattering process. In particular, we first train and validate our qGAN model with artificial data from known underlying probability density functions. Then, in order to test our model in a realistic set-up, we use as training sets simulated Monte Carlo events for particle physics processes at the Large Hadron Collider (LHC) at CERN. (2) We propose an alternative quantum generator architecture. Traditionally, the prior noise distribution, or latent dimension in the language of generative models, is provided to the quantum generator through its first quantum gates. We instead embed it on every layer of the network. This allows us to achieve improved state-of-the-art results with shallow QNNs. In particular, with a binning density that is at least an order of magnitude higher, we achieve significantly smaller Kullback-Leibler (KL) divergences, even when working with a realistic dataset. Note that a similar concept was introduced in the classical context [KLA21], coined as style-based generative adversarial network (GAN), which was proven to be useful in facial recognition tasks. Given this analogy with the classical literature, from now on we refer to our qGAN model as style-qGAN. (3) We validate and assess our style-qGAN in quantum hardware. Specifically, we successfully implement our model in two different quantum architectures, namely ion traps and superconducting qubits.

[ZLW19] Zoufal et al., “Quantum generative adversarial networks for learning and loading random distributions”

[Zen+19] Zeng et al., “Learning and inference on generative adversarial quantum circuits”

[Hu+19] Hu et al., “Quantum generative adversarial learning in a superconducting quantum circuit”

[Ben+19b] Benedetti et al., “Adversarial quantum circuit learning for pure state approximation”

[Sit+20] Situ et al., “Quantum generative adversarial network for generating discrete distribution”

[RAG21] Romero and Aspuru-Guzik, “Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions”

[Niu+21] Niu et al., “Entangling Quantum Generative Adversarial Networks”

[KLA21] Karras et al., “A Style-Based Generator Architecture for Generative Adversarial Networks”

It is important to highlight that several research groups from the high-energy physics (HEP) community are investigating potential applications of quantum technologies in HEP applications and obtaining interesting results [PS+21b; Gua+21; Cha+21a; Cha+21b; Bel+21; Agl+22; AS22; DH22; Del+22]. Therefore, the study presented in this chapter should be considered as proof-of-concept, providing a robust and reproducible starting point for future investigations. In particular, the introduction of GAN models in HEP Monte Carlo simulation has been discussed extensively in the last years, see Refs. [BPW19; BPW20; Bel+20; BP20; Bal+21; Bac+21; But+21]. In this chapter, we consider the possibility to use a qGAN model in a data augmentation context, where the model is trained with a small amount of input samples and it learns how to sample the underlying distribution.

[PS+21b] Pérez-Salinas et al., “Determining the proton content with a quantum computer”

[Gua+21] Guan et al., “Quantum machine learning in high energy physics”

[Cha+21a] Chang et al., “Dual-Parameterized Quantum Circuit GAN Model in High Energy Physics”

[Cha+21b] Chang et al., “Quantum Generative Adversarial Networks in a Continuous-Variable Architecture to Simulate High Energy Physics Detectors”

[Bel+21] Belis et al., “Higgs analysis with quantum classifiers”

[Agl+22] Agliardi et al., “Quantum integration of elementary particle processes”

[AS22] Araz and Spannowsky, “Classical versus Quantum: comparing Tensor Network-based Quantum Circuits on LHC data”

[DH22] Delgado and Hamilton, “Unsupervised Quantum Circuit Learning in High Energy Physics”

[Del+22] Delgado et al., “Quantum Computing for Data Analysis in High-Energy Physics”

6.1 GENERATIVE ADVERSARIAL LEARNING

The classical implementation of a GAN model [Goo+14] involves at least three components: the discriminator model, the generator model, and the adversarial training procedure. Here we consider a hybrid quantum-classical system, where the generator model has a quantum representation through a QNN while the discriminator is a classical neural network model. This choice is motivated by the practical positive implications of a hardware-based generative model, in particular the possibility to obtain performance improvements in a real quantum device. The idea of using a quantum device for the generation of samples is very attractive because the complicated aspects of density modeling and sampling are delegated to a hardware architecture.

There are alternative approaches where both models could be represented by a QNN [DDK18; Hu+19; Ben+19b; RAG21; Niu+21]. However, after testing some prototype architectures, we have observed faster convergence when using a classical discriminator.

[DDK18] Dallaire-Demers and Killoran, "Quantum generative adversarial networks"

[Hu+19] Hu et al., "Quantum generative adversarial learning in a superconducting quantum circuit"

[Ben+19b] Benedetti et al., "Adversarial quantum circuit learning for pure state approximation"

[RAG21] Romero and Aspuru-Guzik, "Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions"

[Niu+21] Niu et al., "Entangling Quantum Generative Adversarial Networks"

In Figure 6.1 we schematically show the steps involved in the style-qGAN presented here. The procedure starts from the preparation of reference samples from a known distribution function that we would like to encode in the quantum generator model. At the same time, we define a QNN model where we inject stochastic noise in the latent space variables; these are used to define all the quantum gates of the network. The generator model is then used to extract fake generated samples that, after the training procedure, should match the quality of the known input distribution. Lastly, both sets of samples are used to train the discriminator model. The quality of the training is measured by an appropriate loss function which is monitored and optimized classically by a minimization algorithm based on the adversarial approach. The training process consists of simultaneous stochastic gradient descent for both models which, after reaching convergence, delivers a quantum generator model with realistic sampling.

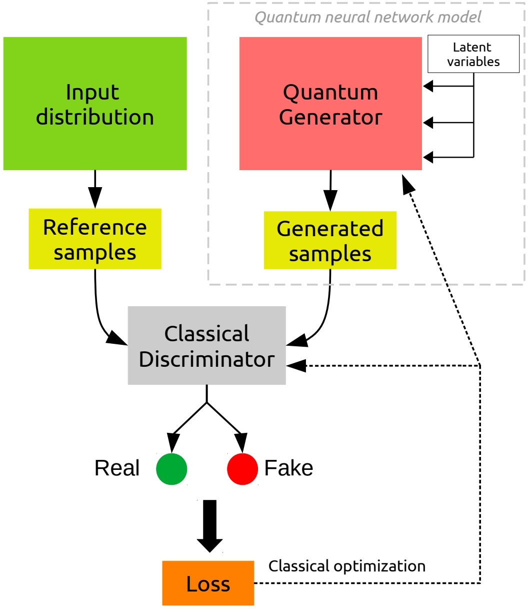

FIGURE 6.1: Schematic steps involved in the style-qGAN training.

In the following paragraphs, we first introduce the optimization procedure and the quantum generator network, and validate the procedure by using reference samples from known distribution functions to train the model on a quantum simulator. We then train the style-qGAN model with Monte Carlo-generated LHC events using again a quantum simulator. Finally, the best-trained model is deployed on real quantum hardware devices based on superconducting and trapped-ion technologies.

All calculations involving quantum circuit simulation are performed using `Qibo` v0.1.6 [Eft+21b; Eft+21a] on classical hardware. For this par-

[Aba+15] Abadi et al., *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*

ticular implementation, we have used the `tensorflow` [Aba+15] backend which provides the possibility to use gradient descent optimizers during the training step. The style-qGAN model is publicly available through the `Qibo` framework and the code repository in [hep+21].

6.2 OPTIMIZATION PROCEDURE

As previously discussed, the style-qGAN comprises of a QNN for the generator $G(\phi_g, z)$ and a classical network for the discriminator $D(\phi_d, x)$, where ϕ_g and ϕ_d are the parameters of the corresponding networks. The quantum generator transforms samples from a prior standard Gaussian noise distribution $z \sim p_{\text{prior}}(z)$, also called latent variables, into samples generated by $G(\phi_g)$, thus mapping $p_{\text{prior}}(z)$ to a different distribution p_{fake} of generated data. On the other hand, the discriminator takes as input samples x and tries to distinguish between fake data from the generator and real data from the reference input distribution p_{real}. The training corresponds to an adversarial game, where we alternate between improving the discriminator to distinguish fake and real data, and the generator to cheat the discriminator with new fake data.

In this implementation, we consider the binary cross-entropy for the optimization objective. The generator's loss function can be defined as

$$\mathcal{L}_G(\phi_g, \phi_d) = -\mathbb{E}_{z\sim p_{\text{prior}}(z)}[\log D(\phi_d, G(\phi_g, z))], \tag{6.1}$$

while the discriminator's loss function can be defined as

$$\begin{aligned} \mathcal{L}_D(\phi_g, \phi_d) = \mathbb{E}_{x\sim p_{\text{real}}(x)}[\log D(\phi_d, x)] \\ + \mathbb{E}_{z\sim p_{\text{prior}}(z)}[\log(1 - D(\phi_d, G(\phi_g, z)))]. \end{aligned} \tag{6.2}$$

Notice that the adversarial training corresponds to a minimax two-player game,

$$\min_{\phi_g} \mathcal{L}_G(\phi_g, \phi_d), \tag{6.3}$$

$$\max_{\phi_d} \mathcal{L}_D(\phi_g, \phi_d), \tag{6.4}$$

where the optimum uniquely corresponds to the Nash equilibrium between the loss functions.

[Zei12] Zeiler, "Adadelta: an adaptive learning rate method"

The optimization of the parameters ϕ_g and ϕ_d is done alternatively by updating the quantum generator and classical discriminator. The optimizer used to update the steps is the ADADELTA [Zei12], which is a stochastic gradient descent method that monotonically decreases its learning rate. The starting learning rates utilized are 0.1 for the classical discriminator and 0.5 for the quantum generator.

6.3 STYLE-BASED QUANTUM GENERATOR ANSATZ

The quantum generator is implemented by a QNN with trainable parameters. In particular, we employ the architecture shown in Figure 6.2. We consider a layered QNN, where each layer is composed of a set of entangling gates U_{ent} preceded by two alternating R_y and R_z single-qubit rotations. After implementing the layered network, a final layer of R_y gates is applied. Note

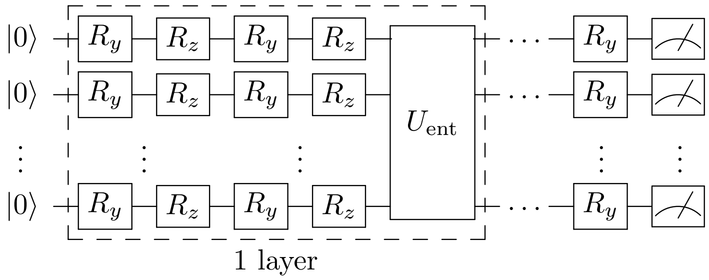

FIGURE 6.2: Quantum neural network employed for the qGAN model. As indicated by the dashed box, each layer is composed of a set of entangling gates U_{ent}, to be specified for each example, preceded by two alternating R_y and R_z single-qubit rotations. After implementing the layered network, a final layer of R_y gates is applied.

that U_{ent} is specific to each example and $R_j(\theta_k) = e^{-i\theta_k\sigma_j/2}$, where σ_j are the Pauli operators. The number of layers can be modified to tune the capacity of the quantum generator. However, in the following we improve upon state-of-the-art results with shallow QNNs that contain one and two layers.

Let us emphasize here the novelty of the quantum generator architecture used for the style-qGAN model, where the action of each qubit rotation is parameterized by the set of trainable parameters $\vec{\phi}_g$ and, most importantly, the latent vector $\vec{\xi}$. Specifically, we encode them by using a linear function as

$$R_{y,z}\left(\vec{\phi}_g, \vec{\xi}\right) = R_{y,z}\left(\phi_g^{(i)}\xi^{(j)} + \phi_g^{(i+1)}\right), \tag{6.5}$$

where i, j indicates the component of the vector. The length of the latent vector $\vec{\xi}$ will depend on the choice of latent dimension D_{latent} for each implementation. Notice that our quantum generator embeds the input latent variables into all the quantum gates of the network, in contrast to previous qGAN proposals. This permits the new architecture to process and decide in which parts of the QNN the latent variables should play a relevant role.

Recall that the quantum generator's task is creating fake samples to fool the classical discriminator. The fake samples are prepared by acting with the parameterized QNN on the initial n-qubit state $|0\rangle^{\otimes n}$, and then measuring in the computational basis. For our implementations, each qubit delivers one sample component. That is, the sample $x \in \mathbb{R}^n$ is generated as

$$x = \left[-\left\langle\sigma_z^1\right\rangle, -\left\langle\sigma_z^2\right\rangle, \ldots, -\left\langle\sigma_z^n\right\rangle\right], \tag{6.6}$$

with

$$\left\langle\sigma_z^i\right\rangle = \left\langle\Psi(\vec{\phi}_g, \vec{\xi})\right|\sigma_z^i\left|\Psi(\vec{\phi}_g, \vec{\xi})\right\rangle, \tag{6.7}$$

where $\left|\Psi(\vec{\phi}_g, \vec{\xi})\right\rangle$ is the output state from the quantum generator. Notice, however, that for other models, more sophisticated ways of generating fake samples could be more convenient to implement. For instance, one could generate a sample component by computing expectation values involving several qubits or generate samples directly from the distribution of computational-basis states. Finally, let us briefly comment that we used a deep convolutional neural network for the discriminator. More details about the classical discriminator implementation can be found in the code [hep+21].

6.4 VALIDATION EXAMPLES

In this section, we show examples of style-qGAN models obtained for known prior distribution functions in one and three dimensions. The results presented here have been obtained after a systematic process of fine-tuning and manual hyper-optimization of the training and quantum generator model.

6.4.1 *1D Gamma distribution*

In order to test the framework proposed above, we consider the sampling of a 1D gamma distribution with probability density function given by

$$p_\gamma(x,\alpha,\beta) = x^{\alpha-1}\frac{e^{-x/\beta}}{\beta^\alpha\Gamma(\alpha)}, \tag{6.8}$$

where Γ is the Gamma function. In this example we take $p_\gamma(x,1,1)$ as the input distribution and train a style-qGAN with 1 qubit, 1 latent dimension and 1 layer using 10^4 samples from the input distribution. The total number of trainable parameters is 10. We perform a linear pre-processing of the data to fit the samples within $x \in [-1,1]$. We undo this transformation after the training. In Figure 6.3 we show the evolution of the loss function for the generator and discriminator models in terms of the number of epochs. We observe the typical behavior of GAN training and a convergence region after 15000 epochs. The style-qGAN is trained with batch sizes of 128 samples.

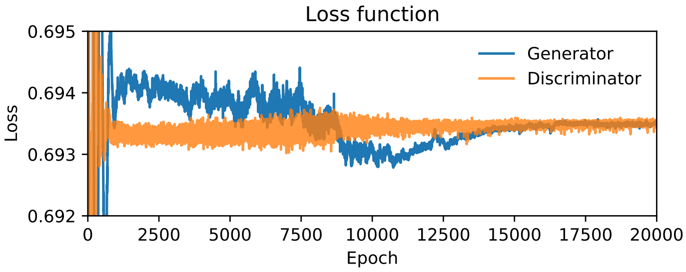

FIGURE 6.3: Example of loss function convergence. After an initial warm-up phase, the loss function of both models converges. This indicates that the style-qGAN has been successfully trained.

A necessary property of this framework is that the style-qGAN model learns the underlying distribution from a small data set. To demonstrate this, we train a style-qGAN model with a set number of reference samples and then use it to generate two sample sets of different size. In particular, we train the style-qGAN with 10^4 reference samples and then use it to generate sets of 10^4 and 10^5 samples.

The top of Figure 6.4 shows the smaller sample distribution generated by the style-qGAN model in blue and a sampling of the reference distribution of the same size in red. This enables a comparison also using the Kullback-Leibler divergence (KL) [KL51]. In both cases, the 10^4 samples have been transformed into histograms with 100 bins linearly spaced on the x-axis of the figure. We observe that the distributions are statistically similar even for this high-density binning choice. The KL divergence of two displayed distributions is 0.141, which entails a high degree of similarity.

[KL51] Kullback and Leibler, “On information and sufficiency”

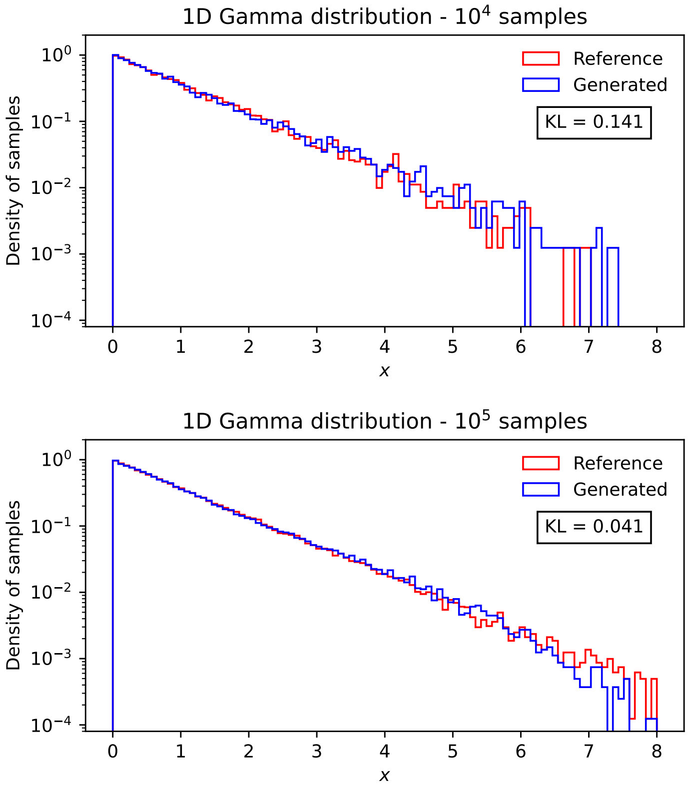

FIGURE 6.4: Examples of 1D gamma distribution sampling for the reference underlying distribution (red) and a style-qGAN model (blue) that has been trained with 10^4 reference samples. The top plot compares 10^4 generated samples. The bottom compares 10^5 samples generated from a style-qGAN model trained with 10^4 reference samples. We observe a good level of agreement between both distributions with low values of the KL distance, despite the model being trained on a small training set.

Going further in the bottom of Figure 6.4 we show the same results as in the top on the larger set containing 10^5 samples. Again, we use for comparison a re-sampling of the reference distribution at the same size as the generated set and show both distributions on a grid with 100 linearly spaced bins. Having generated an order of magnitude more samples than the training set we observe that the style-qGAN model performs well. Both distributions are visually very close to each other and the KL divergence of 0.041 signals a high degree of similarity.

In order to compare the two KL divergences, note that they are computed on discrete histograms. Therefore, for an honest comparison, the number of bins for the larger sample set has to be increased proportionally to the increase in generated sample size, i.e. to compare with 100 bins for the 10^4 sample size we need to set 1000 bins for the 10^5 sample size. In this case, for a style-qGAN that provides an equally good description of the underlying distribution function, the KL divergence will stay constant or decrease. Here, with this change in binning, we find the KL divergences are 0.141 and 0.112, respectively. This behavior confirms that the style-qGAN model is able to learn the underlying distribution function even if trained with a small training sample set. Such a feature is particularly interesting in the context of data augmentation applications [FA+18; TA19], where few samples are available, nonetheless the style-qGAN model can generalize and learn the underlying distribution with satisfactory outcome.

6.4.2 *3D correlated Gaussian distribution*

The previous test shows that a style-qGAN model implemented on a single qubit can be trained and produce acceptable results. However, this particular set-up does not include entanglement between qubits. In order to study the impact of the entanglement term U_{ent} in the considered QNN, we select as an underlying distribution a 3D correlated Gaussian distribution centered at $\overline{\mu} = (0, 0, 0)$ with covariance matrix

$$\sigma = \begin{pmatrix} 0.5 & 0.1 & 0.25 \\ 0.1 & 0.5 & 0.1 \\ 0.25 & 0.1 & 0.5 \end{pmatrix}. \tag{6.9}$$

For this specific set-up, we consider a 3-qubit model with 3 latent dimensions and 1 layer. The U_{ent} consists of two controlled-R_y gates acting sequentially on the 3 qubits. The total number of trainable parameters is 34. As in the previous example, we perform a linear pre-processing of the data to fit the samples within $x \in [-1, 1]$, and then we undo this transformation after the training. In Table 6.1 we summarize the style-qGAN configurations obtained for both examples discussed in this section.

	1D gamma	**3D Gaussian**
Qubits	1	3
D_{latent}	1	3
Layers	1	1
Epochs	3×10^4	1.3×10^4
Training set	10^4	10^4
Batch size	128	128
Parameters	10	34
U_{ent}	None	2 sequential CR_y gates

TABLE 6.1: Summary of the style-qGAN set-up for the 1D gamma distribution and the 3D correlated Gaussian distribution.

Following the same training procedure employed in Section 6.4.1 and again using 10^4 reference samples to train the style-qGAN model, we test how well our model samples this specific 3D correlated Gaussian distribution. The results are shown in Figure 6.5. In the top row, we compare the one-dimensional cumulative projections of samples generated by the style-qGAN model with the reference input distribution function for 10^5 samples. We again use a grid of 100 linearly spaced bins per dimension in order to highlight small differences between the prior reference distribution and the artificial samples. For this example, we also observe that the distributions are statistically similar as the corresponding KL distances are quite small and close to each other. In the second row, we show 10^5 samples produced by the style-qGAN model in two-dimensional projections.

To further study the features of the style-qGAN model in the third row of plots in Figure 6.5 we show the two-dimensional projections of the ratio between samples generated from the prior reference distribution and the style-qGAN model. In this way, we can visualize how well the model learns

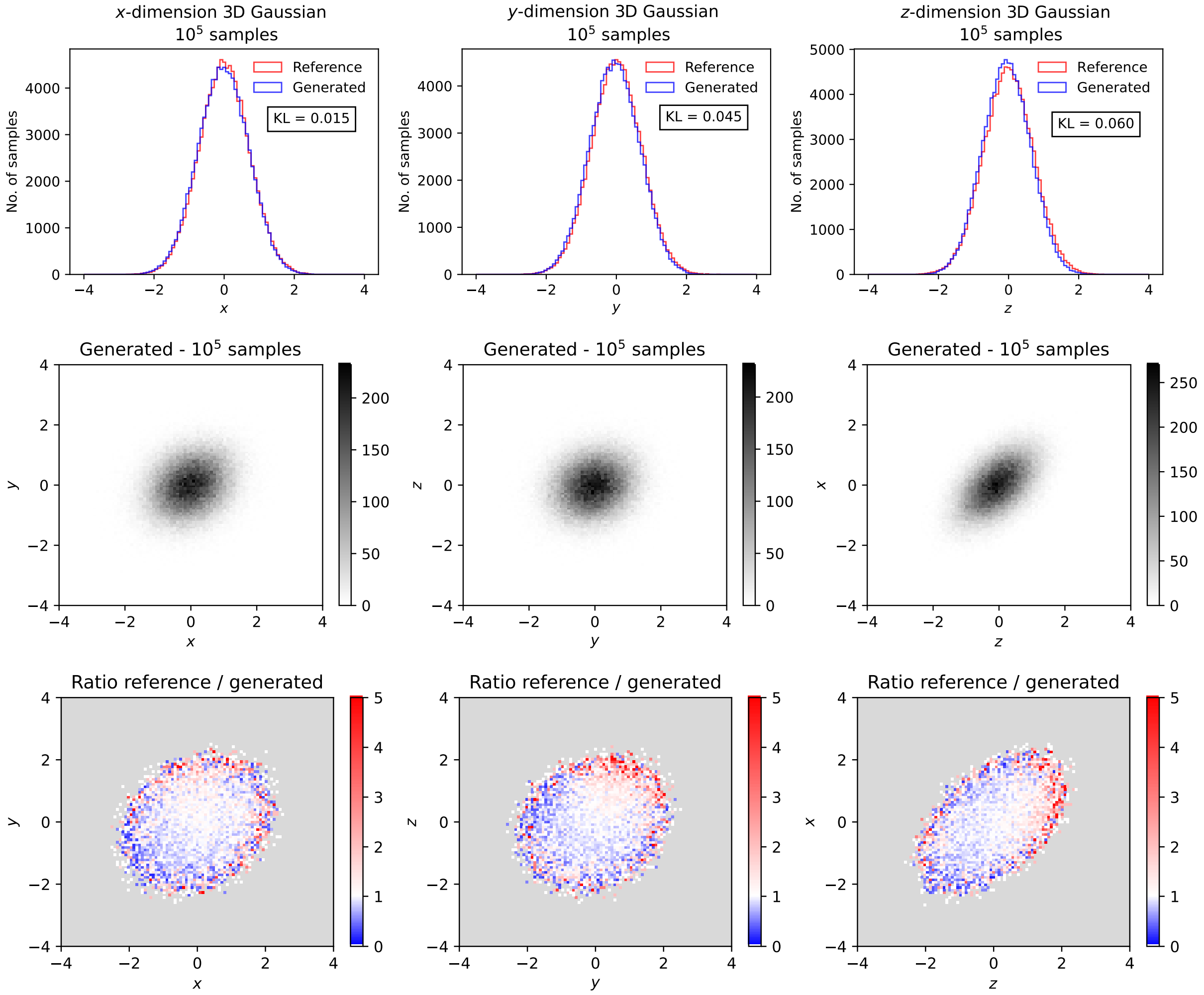

FIGURE 6.5: Marginal samples distributions for each dimension x, y, z of the 3D correlated Gaussian distribution for the style-qGAN model trained with 10^4 samples (top row), together with the corresponding two-dimensional sampling projections (middle row) and the ratio to the reference underlying prior distribution (bottom row). The style-qGAN generator model learns the correlations and provides acceptable samples when compared to the reference distribution. Note that we choose a grey background for the plots at the bottom row to more clearly highlight a ratio of one between reference and generated samples, indicated by white.

not only the distributions but also the correlations between the dimensions of the problem. A ratio of one, given by a white coloring of the corresponding bin in the figure, would imply the reference and generated samples are identical. Note that we aim to generate unseen samples, not an identical copy of the reference set. However, at the same time, the model should not diverge significantly, depicted by deep blue and red in the figure, nor occupy space in the grey area of the figure. We observe a good level of agreement, in particular in those regions where the sampling frequency is higher. The largest deviations are seen at the edges of the distributions, where the sampling frequency is lower. These deviations are evidence of the limitations in our model, common to the GAN method; however, their severe appearance is an artifact of visualization due to data augmentation.

To better quantify how well the correlations have been learned, we study the covariance matrices defined by the reference and the generated samples. The summed eigenvalues of the reference and generated covariance matrices give a means to estimate the similarity between the learned underlying correlations. We find agreement between the reference and generated eigenvalues to the better of 10% for style-qGAN set-ups with equal and more than 3 latent dimensions. Recall that the latent variables are introduced in every gate of the circuit, including the entangling ones U_{ent}. With $D_{\text{latent}} < 3$ we observe significant deviations of factors $\mathcal{O}(10)$ while for $D_{\text{latent}} \geq 3$ no further significant improvement is seen. The same holds for increasing the number of layers in the style-qGAN model. This suggests that the number of latent dimensions introduced is a key hyperparameter once the number of layers allows a sufficient complexity. However, training success also depends on the convergence of the GAN parameters through optimization. This means that, in practice, having more layers and parameters than the minimal set might be a better choice.

Since the eigenvalues are known also exactly through Eq. 6.9 we furthermore can compare the performance of the style-qGAN with increased generation sample size. We find that the style-qGAN with 3 latent dimensions and 1 layer (shown here) generates sets that reproduce the exact eigenvalues of the input covariance matrix to better than $\lesssim 6\%$ for 10^3, $\lesssim 1.3\%$ for 5×10^3 and $\lesssim 0.8\%$ for 2×10^4 samples.

This analysis demonstrates a key property of a functioning GAN model – that the larger set of generated samples more closely agrees with the reference input distribution function. The observation that the style-qGAN fulfils this property confirms its viability as a functioning quantum implementation of the generative adversarial network idea for multi-dimensional correlated data.

6.5 GENERATING LHC EVENTS

After the validation of the style-qGAN model presented in the previous section, let us consider a training dataset from HEP. One of the big challenges involving Monte Carlo (MC) event generation is the large number of statistics required to reconstruct events with high accuracy in order to compare predictions of physical observables to experimental data. Ideally, we could try to learn how a specific physical process generates events.

In this context, we have generated 10^5 MC events for $pp \to t\bar{t}$ production at LHC with $\sqrt{s} = 13$ TeV with MadGraph (`MG5_aMC` [Alw+14; Fre+18]) at leading order in the strong coupling constant. From this simulated events

[Alw+14] Alwall et al., “The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations”

[Fre+18] Frederix et al., “The automation of next-to-leading order electroweak calculations”

	$pp \to t\bar{t}$ **LHC events**
Qubits	3
D_{latent}	5
Layers	2
Epochs	3×10^4
Training set	10^4
Batch size	128
Parameters	62
U_{ent}	2 sequential CR_y gates

TABLE 6.2: Summary of the style-qGAN set-up for the LHC events distribution.

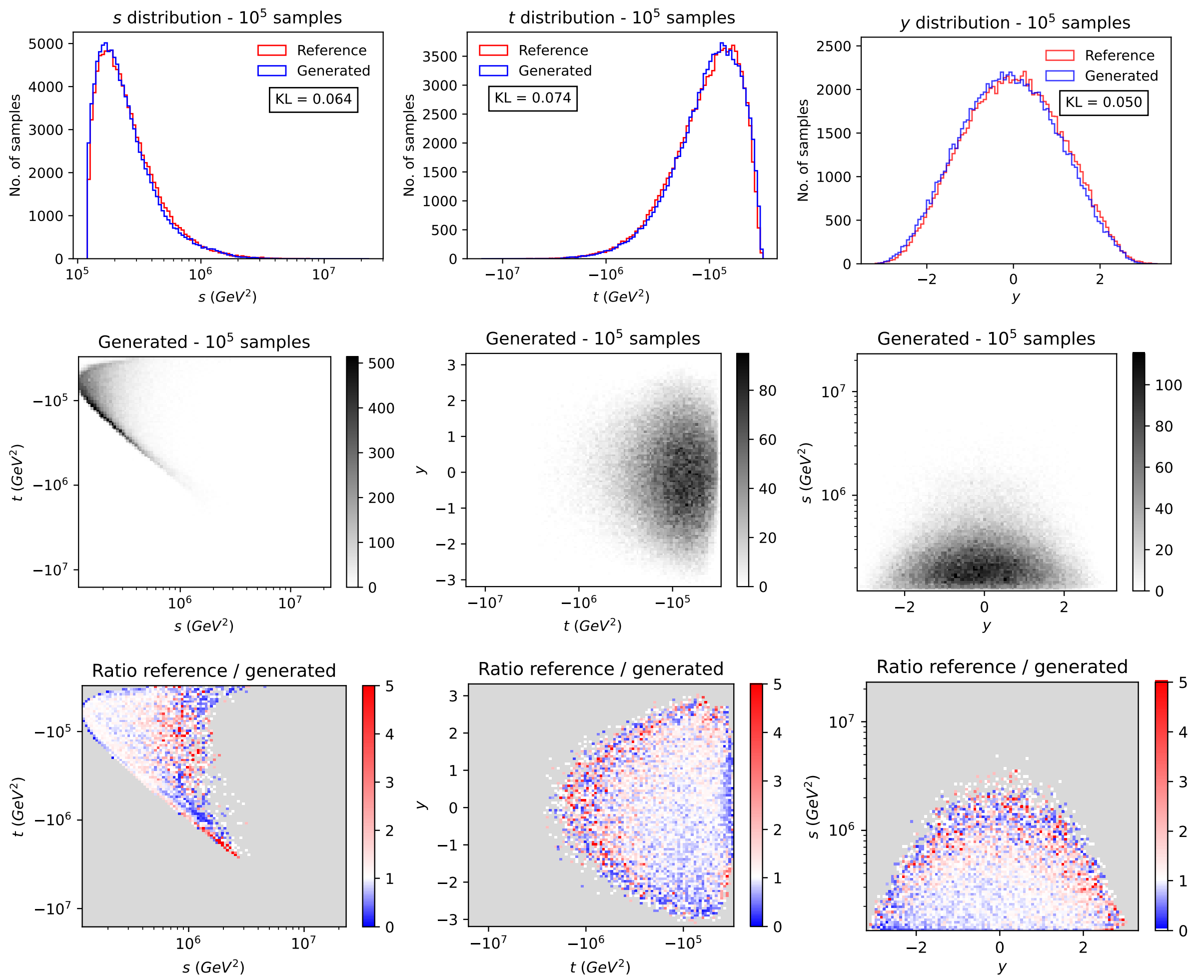

FIGURE 6.6: Marginal samples distributions for the physical observables s, t, y in $pp \rightarrow t\bar{t}$ production at the LHC for the style-qGAN model trained with 10^4 samples (top row), together with the corresponding two-dimensional sampling projections (middle row) and the ratio to the reference underlying prior MC distribution (bottom row). The style-qGAN generator model learns the correlations and provides acceptable samples when compared to the reference distribution. Note that we choose a grey background for the plots at the bottom row to more clearly highlight a ratio of one between reference and generated samples, indicated by white.

we sample the Mandelstam variables (s, t) and the rapidity. Here, s and t are understood as the local partonic variables, $s = (p_1 + p_2)^2$, $t = (p_1 - p_3)^2$, where p_1 and p_2 are the four-momenta of the incoming quarks within the proton that collide to produce a top quark with four-momentum p_3 and an anti-top quark with four-momentum p_4. Note that all momenta are given in the center-of-mass frame.

We consider a 3-qubit model with 5 latent dimensions and 2 layers. Again, U_{ent} consists of two controlled-R_y gates acting sequentially on the 3 qubits. The total number of trainable parameters is 62. The style-qGAN model has been trained on 10^4 samples. See Table 6.2 for more details. In this case, we perform a linear pre-processing of the data to fit the samples within $x \in [-1, 1]$ after a power transform [YJ00] from the Python package `ScikitLearn` [Ped+11]. As previously, we undo this transformation after the training.

Following the same training procedure employed in the previous section, in the top row of Figure 6.6 we compare the one-dimensional cumulative

[YJ00] Yeo and Johnson, "A new family of power transformations to improve normality or symmetry"

[Ped+11] Pedregosa et al., "Scikit-learn: Machine Learning in Python" projections of samples generated by the style-qGAN model with the reference input distribution function for 10^5 samples. We use a grid of 100 linearly spaced bins for y and 100 log-spaced bins for s and t. For this example, the distributions are again statistically similar, with the corresponding KL distance being small and close to each other. In the second row of Figure 6.6 we show 10^5 samples produced by the style-qGAN model in two-dimensional projections.

The bottom row of plots in Figure 6.6 shows the ratio between samples generated from the prior original MC distribution and the style-qGAN model. Again, even for this physically-realistic model, we observe a remarkable level of agreement, especially in those regions where the sampling frequency is higher. Most importantly, we observe that the style-qGAN learns the correlations between the three dimensions.

Applying the same reasoning as in the previous section we compute the eigenvalues of the covariance matrices derived from the reference and generated data sets. To this extent we use the larger sized reference data set calculated previously using MadGraph (`MG5_aMC`). We find that the summed eigenvalues of the covariance matrices derived from samples generated by the shown style-qGAN with 5 latent dimensions and 2 layers agree with the corresponding reference to $\sim 9-13\%$ for 10^3, $\sim 8-15\%$ for 5×10^3 and $\sim 7-14\%$ for 2×10^4 samples. Here the quoted range originates from comparing different samples of the reference data. Furthermore, we suppress effects from the inverse transformation that converts the generated sample and instead focus on the learning capability of the style-qGAN model by estimating the covariances on the transformed reference data sets. It should be stressed that this test is slightly different from the one in the previous section since the exact eigenvalues are not known. As a result, the sampling error of the reference enters and an agreement at the previous level should not be expected as too close of an agreement would indicate the model is overfitted. However, our model exhibits the expected and necessary behaviour, even when applied to real data.

6.6 SAMPLING FROM QUANTUM HARDWARE

In order to benchmark our style-qGAN model on real quantum hardware, we performed several runs on two different types of architectures. This allows us to qualitatively assess the impact of decoherence and noise, issues that are typical for NISQ computers, and to check whether the model can already give good results without waiting for error-corrected machines. The first quantum architecture we used is based on superconducting transmon qubits as provided by IBM Q quantum computers [1]. The second is based on trapped ion technology as provided by IonQ quantum computers [2] and accessible to us using cloud resources from Amazon Web Services (AWS).

Implementing the style-qGAN onto real quantum hardware introduces a new parameter into the model: the number of shots done for each calculation. Specifically, we now perform a quantum experiment each time we measure the three-qubit state, and we collect the results after a set number of experiments (shots) have been carried out. These then build up expectation

---

[1] IBM's roadmap for scaling quantum technology, Sept. 2020.

[2] Scaling IonQ's Quantum Computers: The Roadmap, Dec. 2020.

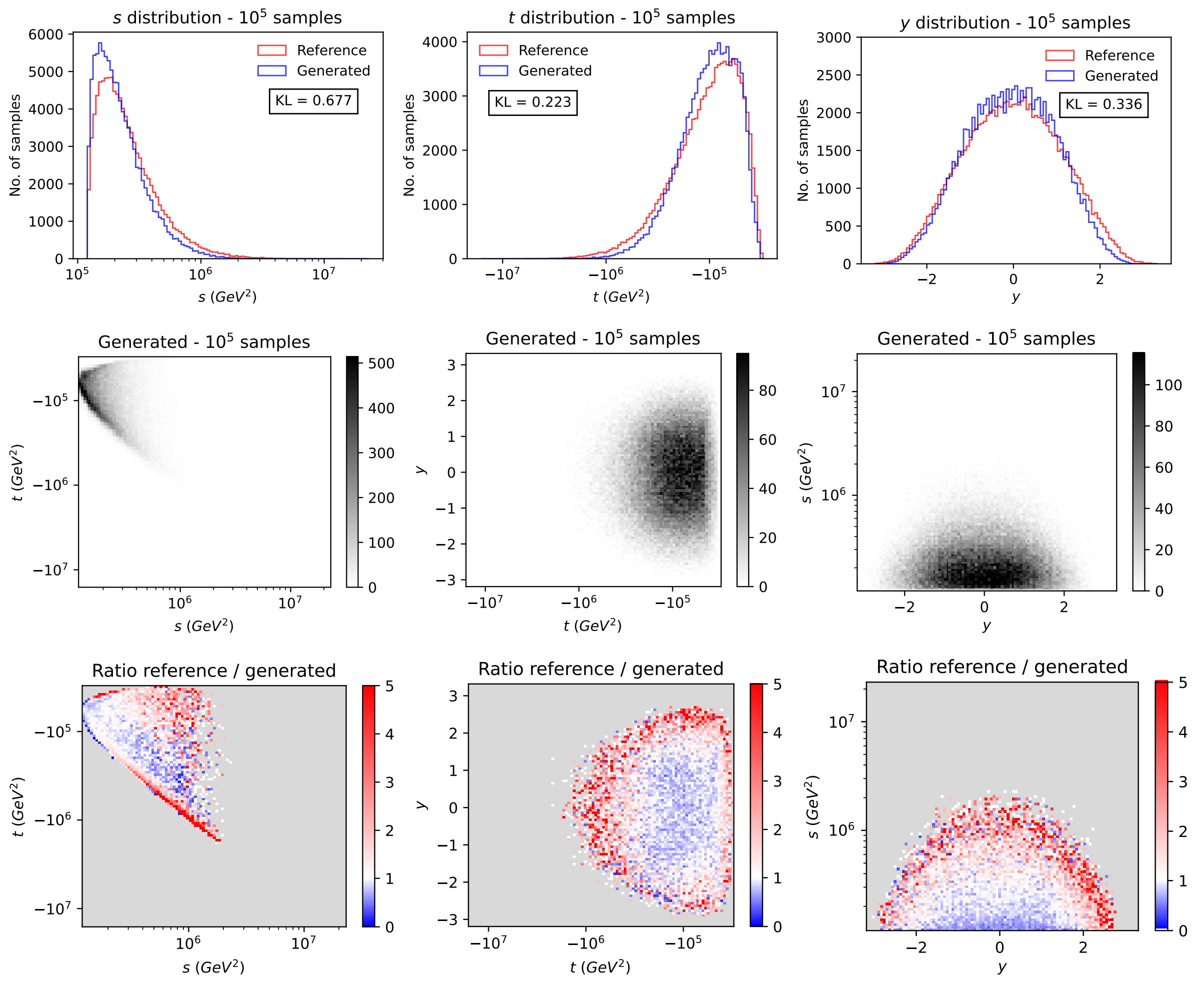

FIGURE 6.7: Marginal samples distributions for the physical observables s, t, y in $pp \to t\bar{t}$ production at the LHC using the style-qGAN generator model trained with 10^4 samples on `ibmq_santiago` (top row), together with the corresponding two-dimensional sampling projections (middle row) and the ratio to the reference underlying prior MC distribution (bottom row). Note that we choose a grey background for the plots at the bottom row to more clearly highlight a ratio of one between reference and generated samples, indicated by white.

values that are used to create generated samples. In this implementation we typically perform a number of 1000 shots per sample.

Prior to running on actual quantum hardware, we performed noise simulations using the IBM Q simplified noise model, which provides an approximation of the properties of real device backends, and enables us to test how well the results presented in Section 6.5 would be preserved in the noisy environment. Results are provided in Appendix D and show that the impact of the noise is expected to be visible to a degree. We leave noise mitigation to further work. For the noise simulation as well as the actual runs on IBM Q quantum devices, we have selected in particular the `ibmq_santiago` 5-qubit Falcon r4L quantum processor. For our circuit, we need only three qubits with at least one directly connected to the other two. We use a translation layer written in `Qiskit` [Ale+19] to implement the circuit in Figure 6.2 and automatically select the three qubits out of the five that have the best noise properties. Note that this also allows us to test the impact of potential interference between qubits, as IonQ qubits are fully

[Ale+19] Aleksandrowicz et al., *Qiskit: An Open-source Framework for Quantum Computing* connected while those of IBM Q are not.

We present in Figure 6.7 examples of samples that have been generated using the `ibmq_santiago` machine on IBM Q. We use a 3-qubit model with 5 latent dimensions and 1 layer and for which the hyperparameters are the same as the ones used in Section 6.5 and trained on 10^4 samples. In contrast to the previous Section 6.5, for this implementation in the quantum hardware we reduced the number of layers to one. This means that we have trained a different style-qGAN with only one layer and then deployed the model to the quantum architecture. This change is motivated by the desire to diminish the effect of noise by reducing the depth of the circuit. Note, the analysis presented in Appendix D shows little deviation between the one- and two-layer result ratios, further strengthening this choice. To compute each fake sample, we have performed 1000 shots on the quantum circuits. In the top row of Figure 6.7, we compare the one-dimensional cumulative projections of samples generated by the style-qGAN model with the reference input distribution functions for 10^5 samples. The binning choice is equivalent to that used in Figure 6.6. In the middle row, we display the generation of 10^5 samples in two-dimensional projections. In the bottom row of plots in Figure 6.7, we show again the ratio between the reference samples, generated using the MC event generator, and the samples generated by the style-qGAN on the `ibmq_santiago` quantum hardware. As expected, the agreement is worse than in Figure 6.6 because of the noise and reduced capacity of the quantum generator, nevertheless the results are reasonable. The style-qGAN generator model deployed in this NISQ hardware still manages to capture the correlations and provides reasonably good samples when compared to the reference distribution. The KL distances reported in the top row of plots are still relatively small, at most one order of magnitude larger than the KL distances reported in Figure 6.6.

During the current NISQ-era, the different quantum hardware architectures are not standardized and can have limits on the potential applications of the machines. As part of the implementation of our model onto quantum hardware, we were also able to study how the style-qGAN performs across different platforms. The aim is to understand whether and to what extent the style-qGAN's performance is hardware-dependent and also its potential hardware transferability. In view of this study of different quantum technologies, we have also performed a run with 10^3 samples only, on IonQ machines and separately on IBM Q machines. We have selected this fairly small amount of samples mainly due to external constraints on IonQ machine access on AWS. Note that the purpose of these tests is not to compare the two different hardware technologies, but instead to test whether the style-qGAN model works well on different quantum architectures. We use again a translation layer, written in Python with the `BraketSDK` from Amazon, between our circuit and the quantum hardware, and we have also performed around 1000 shots for the measurement of the generated samples. We stress again that although the amount of samples is quite low, the purpose of this test is to assess how the algorithm performs on different quantum technologies using the same amount of samples, not to obtain fine-grained results.

We show in Figure 6.8 the two-dimensional projections using IBM Q `ibmq_santiago` machine (upper row) and IBMQ machine (lower row). It

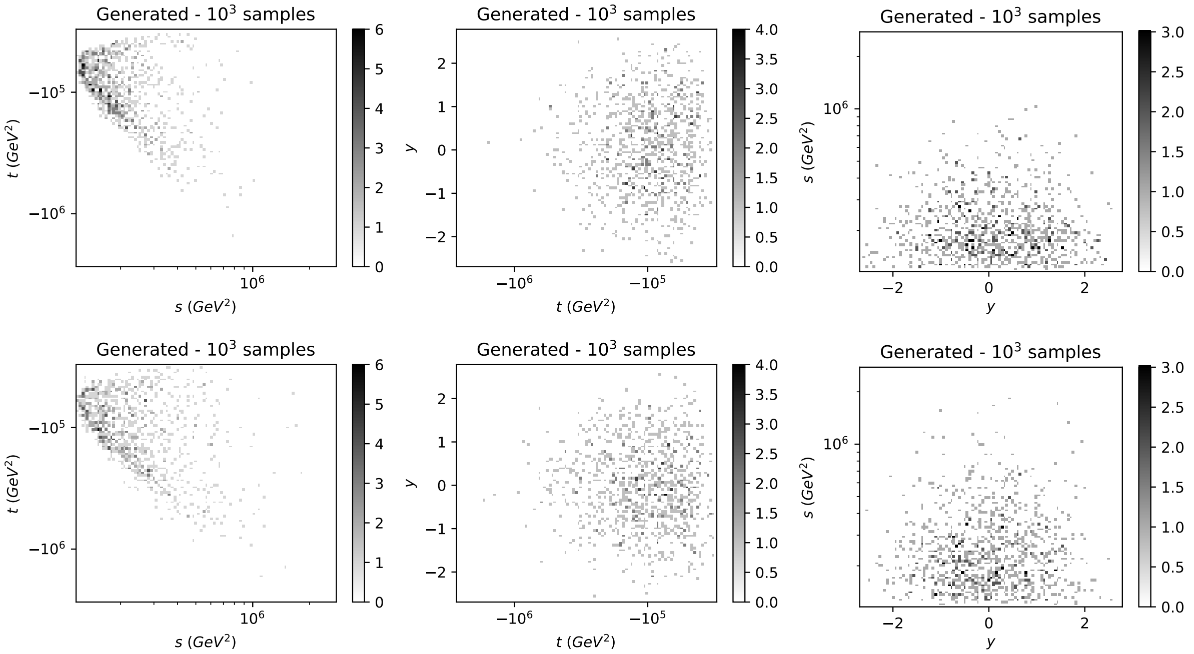

FIGURE 6.8: Example of two-dimensional sampling projections for $pp \to t\bar{t}$ production using the style-qGAN generator model on `ibmq_santiago` (top row) and IonQ (bottom row) trained with 10^4 samples.

is clear that the sampling is sparser than in Figure 6.7, due to the lower number of samples; nonetheless, the style-qGAN captures the underlying distribution and correlations. This is particularly visible on the left plots for the $t - s$ correlation. The comparison between the upper row and the lower row also indicates that both architectures obtain similar results. This demonstrates that the style-qGAN can give good results on two different quantum hardware architectures.

6.7 OUTLOOK

This chapter explores the use of quantum neural networks (QNNs) for Monte Carlo event generation, specifically for scattering processes at the Large Hadron Collider (LHC). We focus specifically on quantum generative adversarial networks (qGANs), which employ two competing networks, the generator and the discriminator, that are trained alternatively. Here we propose a novel quantum generator model that does not follow the traditional path where the prior noise distribution is provided to the quantum generator through its first quantum gates. We instead choose to embed it on every single-qubit and entangling gate of the network. This allows for improvement on state-of-the-art results with a shallow QNN. As a similar concept has been utilized in the classical context, coined as style-GANs, we choose to call our novel architecture a style-qGAN.

As this is a new quantum generative architecture, the body of this work focused on validating and assessing our methodology on various data sets and hardware architectures. In particular, we not only trained our model on toy data, namely 1D gamma and 3D correlated Gaussian distributions, but also on data for real quantum processes at the LHC, generated via MadGraph.

For both toy data and real data, we saw strong evidence that the style-qGAN model could be used for data augmentation, as it was able to reproduce known reference distributions from small sample sets. Additionally, we deployed the models on two different quantum hardware architectures – superconducting qubits (IBM Q) and trapped ions (IonQ). Despite working with a small sample set, we observed that the style-qGAN works well on different hardware architectures. This points to its hardware-independent viability.

The results presented here should be considered as a proof-of-concept, providing a robust and reproducible starting point for future investigations. Nevertheless, this is a first attempt to bridge the power of quantum machine learning algorithms into the complexity of Monte Carlo simulation in HEP. Hopefully, the approach presented here will inspire new HEP applications that may benefit from quantum computing in the future.

Part IV

CONCLUSION

7

Final remarks

“*The end of a melody is not its goal: but nonetheless, had the melody not reached its end it would not have reached its goal either. A parable.*”
—Friedrich Nietzsche

Many of the proposed applications for quantum computing, such as simulating quantum systems or solving large linear systems of equations, are very challenging for current quantum devices, given the limited number of qubits and circuit depth. Variational quantum algorithms (VQAs) have emerged as a leading strategy to address some of these constraints. This thesis has covered several aspects of VQAs, from developing new applications to benchmarks and implementations on real quantum hardware. Chapter 1 is devoted to reviewing the main building blocks of VQAs.

In Chapter 2, we benchmark the accuracy of the variational quantum eigensolver on a finite-depth quantum circuit encoding the ground state of condensed matter systems. For gapped Hamiltonians, we show how the accuracy of the ansatz increases exponentially with the number of layers. For critical systems, however, we observe the appearance of two regimes; a regime that we have called *finite-depth* where the precision of the results only depends on the number of layers and increases very slowly, and the *finite-size* regime where the precision increases exponentially with the number of layers and depends on the system size. This is likely a direct consequence of Lieb-Robinson bounds and the finite speed propagation of the entanglement.

Chapter 3 studies and explores how data encoding influences the behavior of a quantum autoencoder. We show that data encoding helps in the learning task of compressing quantum information. In particular, we compress 1D Ising ground states with higher fidelity by using the same amount of quantum resources as a standard quantum autoencoder. This is, indeed, a step towards what could be done in near-term quantum devices, shortening the distance to practical applications.

In Chapter 4 it has been presented a quantum circuit that produces the elements of a singular value decomposition of pure bipartite states. Its key idea is to demand exact output coincidence on any measurement of the two system parties. We successfully assess the performance of the circuit using random quantum states. Two peculiar spin-offs are as well presented. On the one hand, this circuit can be used to perform the task of a quantum autoencoder. On the other hand, it also can be used to achieve a SWAP operation without any quantum gate that connects the parties.

We continue the study of new applications for VQAs in Chapter 5 presenting the variational quantum linear solver, a VQA for solving linear systems of equations. We provide different cost functions, efficient circuits to estimate them, and an optimization-stopping criterion. On the numerical side, we studied the heuristic scaling of the algorithms by solving non-trivial problems of size up to $2^{50} \times 2^{50}$, and we found scalings at worst linearly in the condition number, logarithmically in the precision, and polylogarithmically in the linear system size. Furthermore, we implemented a particular problem up to a size of 1024×1024 on quantum hardware, which to our knowledge, is the largest implementation at present.

Finally, in Chapter 6, we explore the use of quantum generative adversarial networks for Monte Carlo event generation. We propose a novel quantum generator model that embeds the prior noise distribution on every circuit gate. We validate and assess our generative architecture in toy models, that is, 1D gamma distribution and 3D correlated Gaussian distribution. Then, we train our model on data for real quantum processes at the Large Hadron Collider. For both toy data and real data, we observe strong evidence that the architecture could be used for data augmentation, as it was able to reproduce the reference distributions from a small sample set. Moreover, we successfully deployed the model on two different quantum hardware architecture, namely, superconducting qubits and trapped ions.

The algorithms presented here are examples of potential applications for noisy intermediate-scale quantum computers. Soon, variational quantum algorithms will likely shift from the proposal and development phase to an application phase, with more complex and larger problems being implemented. This thesis paves the way towards new applications and strategies to push the boundaries of near-term quantum devices, with the far-reaching ambition of obtaining quantum advantage.

Bibliography

[BN09] I. Buluta and F. Nori. "Quantum simulators". *Science* **326**, 108–111 (2009) (cit. on pp. 3, 13).

[GAN14] I. M. Georgescu, S. Ashhab, and F. Nori. "Quantum simulation". *Reviews of Modern Physics* **86**, 153 (2014) (cit. on pp. 3, 13).

[Gro97] L. K. Grover. "Quantum mechanics helps in searching for a needle in a haystack". *Physical Review Letters* **79**, 325 (1997) (cit. on p. 3).

[Bra+02] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp. "Quantum amplitude amplification and estimation". *Contemporary Mathematics* **305**, 53–74 (2002) (cit. on p. 3).

[Sho99] P. W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". *SIAM Review* **41**, 303–332 (1999) (cit. on pp. 3, 100).

[Kup05] G. Kuperberg. "A subexponential-time quantum algorithm for the dihedral hidden subgroup problem". *SIAM Journal on Computing* **35**, 170–188 (2005) (cit. on p. 3).

[Hal07] S. Hallgren. "Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem". *Journal of the ACM (JACM)* **54**, 4 (2007) (cit. on p. 3).

[HHL09] A. W. Harrow, A. Hassidim, and S. Lloyd. "Quantum algorithm for linear systems of equations". *Physical Review Letters* **103**, 150502 (2009) (cit. on pp. 3, 45, 54, 55, 57, 58, 60, 63).

[San08] M. Santha. "Quantum walk based search algorithms". *International Conference on Theory and Applications of Models of Computation* **4978**, 31–46 (2008) (cit. on p. 3).

[CVD10] A. M. Childs and W. Van Dam. "Quantum algorithms for algebraic problems". *Reviews of Modern Physics* **82**, 1 (2010) (cit. on p. 3).

[Mon16] A. Montanaro. "Quantum algorithms: an overview". *npj Quantum Information* **2**, 15023 (2016) (cit. on p. 3).

[Fow+12] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland. "Surface codes: Towards practical large-scale quantum computation". *Physical Review A* **86**, 032324 (2012) (cit. on p. 3).

[Bar+14] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, et al. "Superconducting quantum circuits at the surface code threshold for fault tolerance". *Nature* **508**, 500 (2014) (cit. on p. 3).

[Pre18] J. Preskill. "Quantum Computing in the NISQ era and beyond". *Quantum* **2**, 79 (2018) (cit. on pp. 3, 13, 25, 36, 45, 63).

[End+21] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan. "Hybrid quantum-classical algorithms and quantum error mitigation". *Journal of the Physical Society of Japan* **90**, 032001 (2021) (cit. on p. 3).

[Cer+21a] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, et al. "Variational quantum algorithms". *Nature Reviews Physics* **3**, 625–644 (2021) (cit. on pp. 3, 63).

[BK21] L. Bittel and M. Kliesch. "Training variational quantum algorithms is NP-hard". *Physical Review Letters* **127**, 120502 (2021) (cit. on pp. 4, 8).

[Bha+22] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W.-K. Mok, S. Sim, L.-C. Kwek, and A. Aspuru-Guzik. "Noisy intermediate-scale quantum algorithms". *Reviews of Modern Physics* **94**, 015004 (2022) (cit. on pp. 5, 63).

[Bia21] J. Biamonte. "Universal variational quantum computation". *Physical Review A* **103**, L030401 (2021) (cit. on p. 5).

[DN06] C. M. Dawson and M. A. Nielsen. "The Solovay-Kitaev Algorithm". *Quantum Information and Computation* **6**, 81–95 (2006) (cit. on p. 5).

[NC11] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press. (2011) (cit. on pp. 5, 50).

[Kit+02] A. Y. Kitaev, A. Shen, M. N. Vyalyi, and M. N. Vyalyi. *Classical and Quantum Computation*. American Mathematical Society. (2002) (cit. on pp. 5, 40).

[HRC02] A. W. Harrow, B. Recht, and I. L. Chuang. "Efficient discrete approximations of quantum gates". *Journal of Mathematical Physics* **43**, 4445–4451 (2002) (cit. on pp. 5, 40).

[Kan+17] 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) (cit. on pp. 6, 53).

[SJAG19] S. Sim, P. D. Johnson, and A. Aspuru-Guzik. "Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms". *Advanced Quantum Technologies* **2**, 1900070 (2019) (cit. on pp. 6, 63).

[Woi+20] A. J. Woitzik, P. K. Barkoutsos, F. Wudarski, A. Buchleitner, and I. Tavernelli. "Entanglement production and convergence properties of the variational quantum eigensolver". *Physical Review A* **102**, 042402 (2020) (cit. on p. 6).

[BP+20] C. Bravo-Prieto, J. Lumbreras-Zarapico, L. Tagliacozzo, and J. I. Latorre. "Scaling of variational quantum circuit depth for condensed matter systems". *Quantum* **4**, 272 (2020) (cit. on pp. 6, 63).

[Bar+18] P. K. Barkoutsos, J. F. Gonthier, I. Sokolov, N. Moll, G. Salis, A. Fuhrer, M. Ganzhorn, D. J. Egger, M. Troyer, A. Mezzacapo, et al. "Quantum algorithms for electronic structure calculations: Particle-hole Hamiltonian and optimized wave-function expansions". *Physical Review A* **98**, 022322 (2018) (cit. on p. 6).

[Gan+19] M. Ganzhorn, D. J. Egger, P Barkoutsos, P. Ollitrault, G. Salis, N. Moll, M Roth, A Fuhrer, P Mueller, S Woerner, et al. "Gate-efficient simulation of molecular eigenstates on a quantum computer". *Physical Review Applied* **11**, 044092 (2019) (cit. on p. 6).

[Gar+20] B. T. Gard, L. Zhu, G. S. Barron, N. J. Mayhall, S. E. Economou, and E. Barnes. "Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm". *npj Quantum Information* **6**, 1–9 (2020) (cit. on p. 6).

[Bar+21] G. S. Barron, B. T. Gard, O. J. Altman, N. J. Mayhall, E. Barnes, and S. E. Economou. "Preserving symmetries for variational quantum eigensolvers in the presence of noise". *Physical Review Applied* **16**, 034003 (2021) (cit. on p. 6).

[HS88] M. R. Hoffmann and J. Simons. "A unitary multiconfigurational coupled-cluster method: Theory and applications". *The Journal of Chemical Physics* **88**, 993–1002 (1988) (cit. on p. 6).

[TB06] A. G. Taube and R. J. Bartlett. "New perspectives on unitary coupled-cluster theory". *International Journal of Quantum Chemistry* **106**, 3393–3401 (2006) (cit. on p. 6).

[McC+16] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik. "The theory of variational hybrid quantum-classical algorithms". *New Journal of Physics* **18**, 023023 (2016) (cit. on pp. 6, 8).

[Rom+18] 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 Science and Technology* **4**, 014008 (2018) (cit. on p. 6).

[Oll+20] P. J. Ollitrault, A. Baiardi, M. Reiher, and I. Tavernelli. "Hardware efficient quantum algorithms for vibrational structure calculations". *Chemical Science* **11**, 6842–6855 (2020) (cit. on p. 6).

[Miz+20] W. Mizukami, K. Mitarai, Y. O. Nakagawa, T. Yamamoto, T. Yan, and Y.-y. Ohnishi. "Orbital optimized unitary coupled cluster theory for quantum computer". *Physical Review Research* **2**, 033421 (2020) (cit. on p. 6).

[Wan+21a] Q. Wang, M. Li, C. Monroe, and Y. Nam. "Resource-Optimized Fermionic Local-Hamiltonian Simulation on a Quantum Computer for Quantum Chemistry". *Quantum* **5**, 509 (2021) (cit. on p. 6).

[FGG14] E. Farhi, J. Goldstone, and S. Gutmann. "A quantum approximate optimization algorithm". *arXiv preprint arXiv:1411.4028* (2014) (cit. on pp. 6, 16, 53, 54, 100).

[Had+19] S. Hadfield, Z. Wang, B. O'gorman, E. G. Rieffel, D. Venturelli, and R. Biswas. "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz". *Algorithms* **12**, 34 (2019) (cit. on pp. 6, 53).

[Suz76] M. Suzuki. "Generalized Trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems". *Communications in Mathematical Physics* **51**, 183–190 (1976) (cit. on p. 7).

[Far+00] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. "Quantum computation by adiabatic evolution". *arXiv preprint quant-ph/0001106* (2000) (cit. on p. 7).

[Gri+19] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall. "An adaptive variational algorithm for exact molecular simulations on a quantum computer". *Nature Communications* **10**, 1–9 (2019) (cit. on p. 7).

[Rat+19] A. G. Rattew, S. Hu, M. Pistoia, R. Chen, and S. Wood. "A domain-agnostic, noise-resistant, hardware-efficient evolutionary variational quantum eigensolver". *arXiv preprint arXiv:1910.09694* (2019) (cit. on p. 7).

[Bil+21] M Bilkis, M Cerezo, G. Verdon, P. J. Coles, and L. Cincio. "A semi-agnostic ansatz with variable structure for quantum machine learning". *arXiv preprint arXiv:2103.06712* (2021) (cit. on p. 7).

[Tan+21] H. L. Tang, V. Shkolnikov, G. S. Barron, H. R. Grimsley, N. J. Mayhall, E. Barnes, and S. E. Economou. "Qubit-ADAPT-VQE: An adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor". *PRX Quantum* **2**, 020310 (2021) (cit. on p. 7).

[Kha+19b] S. Khatri, R. LaRose, A. Poremba, L. Cincio, A. T. Sornborger, and P. J. Coles. "Quantum-assisted quantum compiling". *Quantum* **3**, 140 (2019) (cit. on pp. 7, 8, 46).

[MF19] K. Mitarai and K. Fujii. "Methodology for replacing indirect measurements with direct measurements". *Physical Review Research* **1**, 013006 (2019) (cit. on p. 8).

[AJL09] D. Aharonov, V. Jones, and Z. Landau. "A polynomial quantum algorithm for approximating the Jones polynomial". *Algorithmica* **55**, 395–421 (2009) (cit. on p. 8).

[Buh+01] H. Buhrman, R. Cleve, J. Watrous, and R. De Wolf. "Quantum fingerprinting". *Physical Review Letters* **87**, 167902 (2001) (cit. on p. 8).

[GC01] D. Gottesman and I. Chuang. "Quantum digital signatures". *arXiv preprint quant-ph/0105032* (2001) (cit. on p. 8).

[Cin+18] L. Cincio, Y. Subaşı, A. T. Sornborger, and P. J. Coles. "Learning the quantum algorithm for state overlap". *New Journal of Physics* **20**, 113022 (2018) (cit. on pp. 8, 51–54).

[BP+19] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. J. Coles. "Variational quantum linear solver". *arXiv preprint arXiv:1909.05820* (2019) (cit. on p. 8).

[McC+18] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven. "Barren plateaus in quantum neural network training landscapes". *Nature Communications* **9**, 1–6 (2018) (cit. on pp. 8, 28, 40, 53).

[Wan+21b] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. J. Coles. "Noise-induced barren plateaus in variational quantum algorithms". *Nature Communications* **12**, 1–11 (2021) (cit. on p. 9).

[MKW21] C. O. Marrero, M. Kieferová, and N. Wiebe. "Entanglement-induced barren plateaus". *PRX Quantum* **2**, 040316 (2021) (cit. on p. 9).

[Cer+21b] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles. "Cost function dependent barren plateaus in shallow parametrized quantum circuits". *Nature Communications* **12**, 1–12 (2021) (cit. on pp. 9, 28, 40, 48, 53).

[UB21] A. Uvarov and J. D. Biamonte. "On barren plateaus and cost function locality in variational quantum algorithms". *Journal of Physics A: Mathematical and Theoretical* **54**, 245301 (2021) (cit. on p. 9).

[GSL18] A Garcia-Saez and J. I. Latorre. "Addressing hard classical problems with adiabatically assisted variational quantum eigensolvers". *arXiv preprint arXiv:1806.02287* (2018) (cit. on pp. 9, 18, 40).

[Gra+19] E. Grant, L. Wossnig, M. Ostaszewski, and M. Benedetti. "An initialization strategy for addressing barren plateaus in parametrized quantum circuits". *Quantum* **3**, 214 (2019) (cit. on pp. 9, 40, 53).

[CLKAG21] A. Cervera-Lierta, J. S. Kottmann, and A. Aspuru-Guzik. "Meta-variational quantum eigensolver: Learning energy profiles of parameterized hamiltonians for quantum simulation". *PRX Quantum* **2**, 020329 (2021) (cit. on pp. 9, 26, 29).

[Zha+22] K. Zhang, M.-H. Hsieh, L. Liu, and D. Tao. "Gaussian initializations help deep variational quantum circuits escape from the barren plateau". *arXiv preprint arXiv:2203.09376* (2022) (cit. on p. 9).

[VC21] T. Volkoff and P. J. Coles. "Large gradients via correlation in random parameterized quantum circuits". *Quantum Science and Technology* **6**, 025008 (2021) (cit. on pp. 9, 53).

[Ama+22] D. Amaro, C. Modica, M. Rosenkranz, M. Fiorentini, M. Benedetti, and M. Lubasch. "Filtering variational quantum algorithms for combinatorial optimization". *Quantum Science and Technology* **7**, 015021 (2022) (cit. on p. 9).

[Sha+20a] K. Sharma, M. Cerezo, L. Cincio, and P. J. Coles. "Trainability of dissipative perceptron-based quantum neural networks". *arXiv preprint arXiv:2005.12458* (2020) (cit. on p. 9).

[Wie+20] R. Wiersema, C. Zhou, Y. de Sereville, J. F. Carrasquilla, Y. B. Kim, and H. Yuen. "Exploring entanglement and optimization within the hamiltonian variational ansatz". *PRX Quantum* **1**, 020319 (2020) (cit. on p. 9).

[Pes+21] A. Pesah, M Cerezo, S. Wang, T. Volkoff, A. T. Sornborger, and P. J. Coles. "Absence of barren plateaus in quantum convolutional neural networks". *Physical Review X* **11**, 041011 (2021) (cit. on p. 9).

[Hol+22] Z. Holmes, K. Sharma, M Cerezo, and P. J. Coles. "Connecting ansatz expressibility to gradient magnitudes and barren plateaus". *PRX Quantum* **3**, 010313 (2022) (cit. on p. 9).

[KB14] D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization". *arXiv preprint arXiv:1412.6980* (2014) (cit. on p. 9).

[Zei12] M. D. Zeiler. "Adadelta: an adaptive learning rate method". *arXiv preprint arXiv:1212.5701* (2012) (cit. on pp. 9, 66).

[DHS11] J. Duchi, E. Hazan, and Y. Singer. "Adaptive subgradient methods for online learning and stochastic optimization". *Journal of Machine Learning Research* **12**, (2011) (cit. on p. 9).

[Küb+20] J. M. Kübler, A. Arrasmith, L. Cincio, and P. J. Coles. "An adaptive optimizer for measurement-frugal variational algorithms". *Quantum* **4**, 263 (2020) (cit. on pp. 9, 40).

[Swe+20] R. Sweke, F. Wilde, J. Meyer, M. Schuld, P. K. Fährmann, B. Meynard-Piganeau, and J. Eisert. "Stochastic gradient descent for hybrid quantum-classical optimization". *Quantum* **4**, 314 (2020) (cit. on p. 9).

[McA+19] S. McArdle, T. Jones, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan. "Variational ansatz-based quantum simulation of imaginary time evolution". *npj Quantum Information* **5**, 1–6 (2019) (cit. on p. 9).

[Sto+20] J. Stokes, J. Izaac, N. Killoran, and G. Carleo. "Quantum natural gradient". *Quantum* **4**, 269 (2020) (cit. on p. 9).

[KB19] B. Koczor and S. C. Benjamin. "Quantum natural gradient generalised to non-unitary circuits". *arXiv preprint arXiv:1912.08660* (2019) (cit. on p. 9).

[BS02] H.-G. Beyer and H.-P. Schwefel. "Evolution strategies–a comprehensive introduction". *Natural Computing* **1**, 3–52 (2002) (cit. on p. 9).

[Zha+20] T. Zhao, G. Carleo, J. Stokes, and S. Veerapaneni. "Natural evolution strategies and variational Monte Carlo". *Machine Learning: Science and Technology* **2**, 02LT01 (2020) (cit. on p. 9).

[ADAG21] A. Anand, M. Degroote, and A. Aspuru-Guzik. "Natural evolutionary strategies for variational quantum computation". *Machine Learning: Science and Technology* **2**, 045012 (2021) (cit. on p. 9).

[GSR19] A. Garcia-Saez and J. Riu. "Quantum observables for continuous control of the quantum approximate optimization algorithm via reinforcement learning". *arXiv preprint arXiv:1911.09682* (2019) (cit. on p. 9).

[Kha+19a] S. Khairy, R. Shaydulin, L. Cincio, Y. Alexeev, and P. Balaprakash. "Reinforcement-learning-based variational quantum circuits optimization for combinatorial problems". *arXiv preprint arXiv:1911.04574* (2019) (cit. on p. 9).

[Wau+20] M. M. Wauters, E. Panizon, G. B. Mbeng, and G. E. Santoro. "Reinforcement-learning-assisted quantum optimization". *Physical Review Research* **2**, 033446 (2020) (cit. on p. 9).

[YBL20] J. Yao, M. Bukov, and L. Lin. "Policy gradient based quantum approximate optimization algorithm". *Mathematical and Scientific Machine Learning*, 605–634 (2020) (cit. on p. 9).

[HM21] D. A. Herrera-Martí. "Policy Gradient Approach to Compilation of Variational Quantum Circuits". *arXiv preprint arXiv:2111.10227* (2021) (cit. on p. 9).

[Spa+92] J. C. Spall et al. "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation". *IEEE Transactions on Automatic Control* **37**, 332–341 (1992) (cit. on p. 9).

[BMK10] K. L. Brown, W. J. Munro, and V. M. Kendon. "Using quantum computers for quantum simulation". *Entropy* **12**, 2268–2307 (2010) (cit. on p. 13).

[AL99] D. S. Abrams and S. Lloyd. "Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectors". *Physical Review Letters* **83**, 5162–5165 (1999) (cit. on p. 13).

[VCL09] F. Verstraete, J. I. Cirac, and J. I. Latorre. "Quantum circuits for strongly correlated quantum systems". *Physical Review A* **79**, 032316 (2009) (cit. on p. 13).

[Tem+11] K. Temme, T. J. Osborne, K. G. Vollbrecht, D. Poulin, and F. Verstraete. "Quantum Metropolis sampling". *Nature* **471**, 87–90 (2011) (cit. on p. 13).

[JLP12] S. P. Jordan, K. S. M. Lee, and J. Preskill. "Quantum Algorithms for Quantum Field Theories". *Science* **336**, 1130 (2012) (cit. on p. 13).

[Ber+18] D. W. Berry, M. Kieferová, A. Scherer, Y. R. Sanders, G. H. Low, N. Wiebe, C. Gidney, and R. Babbush. "Improved techniques for preparing eigenstates of fermionic Hamiltonians". *npj Quantum Information* **4**, 22 (2018) (cit. on p. 13).

[Cao+19] 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, S. Sim, L. Veis, and A. Aspuru-Guzik. "Quantum Chemistry in the Age of Quantum Computing". *Chemical Reviews* **119**, 10856–10915 (2019) (cit. on pp. 13, 46).

[Per+14] 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". *Nature Communications* **5**, 4213 (2014) (cit. on pp. 13, 25, 46, 100).

[Til+21] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, et al. "The Variational Quantum Eigensolver: a review of methods and best practices". *arXiv preprint arXiv:2111.05176* (2021) (cit. on p. 13).

[Bau+12] A. Baumgärtner, A. Burkitt, D. Ceperley, H De Raedt, A. Ferrenberg, D. Heermann, H. Herrmann, D. Landau, D Levesque, W von der Linden, et al. *The Monte Carlo method in condensed matter physics*. Springer Science and Business Media. (2012) (cit. on p. 14).

[Orú14] R. Orús. "A practical introduction to tensor networks: Matrix product states and projected entangled pair states". *Annals of physics* **349**, 117–158 (2014) (cit. on p. 14).

[LK75] J. K. Lenstra and A. R. Kan. "Some simple applications of the travelling salesman problem". *Journal of the Operational Research Society* **26**, 717–733 (1975) (cit. on p. 14).

[Tov84] C. A. Tovey. "A simplified NP-complete satisfiability problem". *Discrete Applied Mathematics* **8**, 85–89 (1984) (cit. on p. 14).

[Osb12] T. J. Osborne. "Hamiltonian complexity". *Reports on Progress in Physics* **75**, 022001 (2012) (cit. on p. 14).

[GW93] S. D. Głazek and K. G. Wilson. "Renormalization of Hamiltonians". *Physical Review D* **48**, 5863–5872 (1993) (cit. on pp. 14, 23).

[Weg94] F. Wegner. "Flow-equations for Hamiltonians". *Annalen der Physik* **506**, 77–91 (1994) (cit. on pp. 14, 23).

[Gla94] S. D. Glazek. "Perturbative renormalization group for Hamiltonians". *Physical Review D* **49**, 4214–4218 (1994) (cit. on pp. 14, 23).

[DU04] S. Dusuel and G. S. Uhrig. "The Quartic Oscillator: a Non-Perturbative Study by Continuous Unitary Transformations". *Journal of Physics A: Mathematical and General* **37**, (2004) (cit. on pp. 14, 18, 23).

[HW05] M. B. Hastings and X.-G. Wen. "Quasiadiabatic continuation of quantum states: The stability of topological ground-state degeneracy and emergent gauge invariance". *Physical Review B* **72**, 045141 (2005) (cit. on pp. 14, 23).

[HC15] Y. Huang and X. Chen. "Quantum circuit complexity of one-dimensional topological phases". *Physical Review B* **91**, 195143 (2015) (cit. on p. 14).

[Cir+17] J. I. Cirac, D. Perez-Garcia, N. Schuch, and F. Verstraete. "Matrix product unitaries: structure, symmetries, and topological invariants". *Journal of Statistical Mechanics: Theory and Experiment* **2017**, 083105 (2017) (cit. on pp. 14, 16, 19).

[KLP18] P. Kos, M. Ljubotina, and T. Prosen. "Many-Body Quantum Chaos: Analytic Connection to Random Matrix Theory". *Physical Review X* **8**, 021062 (2018) (cit. on p. 14).

[LR72] E. H. Lieb and D. W. Robinson. "The finite group velocity of quantum spin systems". *Communications in Mathematical Physics* **28**, 251–257 (1972) (cit. on pp. 16, 24).

[Has07] M. B. Hastings. "An area law for one-dimensional quantum systems". *Journal of Statistical Mechanics: Theory and Experiment* **2007**, P08024 (2007) (cit. on p. 16).

[ECP10] J. Eisert, M. Cramer, and M. B. Plenio. "Colloquium: Area laws for the entanglement entropy". *Reviews of Modern Physics* **82**, 277–306 (2010) (cit. on p. 16).

[Laf16] N. Laflorencie. "Quantum entanglement in condensed matter systems". *Physics Reports*. Quantum entanglement in condensed matter systems **646**, 1–59 (2016) (cit. on p. 16).

[Aha+09] D. Aharonov, D. Gottesman, S. Irani, and J. Kempe. "The power of quantum systems on a line". *Communications in Mathematical Physics* **287**, 41–65 (2009) (cit. on p. 16).

[HLW94] C. Holzhey, F. Larsen, and F. Wilczek. "Geometric and renormalized entropy in conformal field theory". *Nuclear Physics B* **424**, 443–467 (1994) (cit. on p. 16).

[CW94] C. Callan and F. Wilczek. "On geometric entropy". *Physics Letters B* **333**, 55–61 (1994) (cit. on p. 16).

[LRV04] J. I. Latorre, E. Rico, and G. Vidal. "Ground state entanglement in quantum spin chains". *Quantum Information and Computation* **4**, 48–92 (2004) (cit. on p. 16).

[CC04] P. Calabrese and J. Cardy. "Entanglement entropy and quantum field theory". *Journal of Statistical Mechanics: Theory and Experiment* **2004**, P06002 (2004) (cit. on p. 16).

[MFS19] G. B. Mbeng, R. Fazio, and G. Santoro. "Quantum annealing: A journey through digitalization, control, and hybrid quantum variational schemes". *arXiv preprint arXiv:1906.08948* (2019) (cit. on p. 16).

[SML64] T. D. Schultz, D. C. Mattis, and E. H. Lieb. "Two-Dimensional Ising Model as a Soluble Problem of Many Fermions". *Reviews of Modern Physics* **36**, 856–871 (1964) (cit. on p. 17).

[BPZ84] A. A. Belavin, A. M. Polyakov, and A. B. Zamolodchikov. "Infinite conformal symmetry of critical fluctuations in two dimensions". *Journal of Statistical Physics* **34**, 763–774 (1984) (cit. on p. 17).

[Hen99] M. Henkel. *Conformal Invariance and Critical Phenomena*. Theoretical and Mathematical Physics. Berlin Heidelberg: Springer-Verlag. (1999) (cit. on p. 17).

[Ess+05] F. H. L. Essler, H. Frahm, F. Göhmann, A. Klümper, and V. E. Korepin. *The One-Dimensional Hubbard Model*. Cambridge University Press. (2005) (cit. on p. 17).

[Byr+95] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. "A Limited Memory Algorithm for Bound Constrained Optimization". *SIAM Journal on Scientific Computing* **16**, 1190–1208 (1995) (cit. on p. 18).

[Vir+20] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al. "SciPy 1.0: fundamental algorithms for scientific computing in Python". *Nature Methods* **17**, 261–272 (2020) (cit. on pp. 18, 41).

[JNN13] J. R. Johansson, P. D. Nation, and F. Nori. "QuTiP 2: A Python framework for the dynamics of open quantum systems". *Computer Physics Communications* **184**, 1234–1240 (2013) (cit. on p. 18).

[Aff86] I. Affleck. "Universal term in the free energy at a critical point and the conformal anomaly". *Physical Review Letters* **56**, 746–748 (1986) (cit. on p. 20).

[Car86] J. L. Cardy. "Operator content of two-dimensional conformally invariant theories". *Nuclear Physics B* **270**, 186–204 (1986) (cit. on p. 20).

[Tag+08] L. Tagliacozzo, T. R. de Oliveira, S. Iblisdir, and J. I. Latorre. "Scaling of entanglement support for matrix product states". *Physical Review B* **78**, 024410 (2008) (cit. on pp. 20, 22).

[Pol+09] F. Pollmann, S. Mukerjee, A. M. Turner, and J. E. Moore. "Theory of Finite-Entanglement Scaling at One-Dimensional Quantum Critical Points". *Physical Review Letters* **102**, 255701 (2009) (cit. on p. 20).

[Pir+12] B. Pirvu, G. Vidal, F. Verstraete, and L. Tagliacozzo. "Matrix product states for critical spin chains: Finite-size versus finite-entanglement scaling". *Physical Review B* **86**, 075117 (2012) (cit. on p. 20).

[Sto+15] V. Stojevic, J. Haegeman, I. P. McCulloch, L. Tagliacozzo, and F. Verstraete. "Conformal data from finite entanglement scaling". *Physical Review B* **91**, 035120 (2015) (cit. on p. 20).

[Van+17] L. Vanderstraeten, M. Mariën, J. Haegeman, N. Schuch, J. Vidal, and F. Verstraete. "Bridging Perturbative Expansions with Tensor Networks". *Physical Review Letters* **119**, 070401 (2017) (cit. on p. 23).

[BHV06] S. Bravyi, M. B. Hastings, and F. Verstraete. "Lieb-Robinson Bounds and the Generation of Correlations and Topological Quantum Order". *Physical Review Letters* **97**, 050401 (2006) (cit. on p. 24).

[Con+21] M. Consiglio, W. J. Chetcuti, C. Bravo-Prieto, S. Ramos-Calderer, A. Minguzzi, J. I. Latorre, L. Amico, and T. J. Apollaro. "Variational Quantum Eigensolver for SU(N) Fermions". *arXiv preprint arXiv:2106.15552* (2021) (cit. on p. 24).

[JSP22] B. Jobst, A. Smith, and F. Pollmann. "Finite-depth scaling of infinite quantum circuits for quantum critical points". *arXiv preprint arXiv:2203.11975* (2022) (cit. on p. 24).

[Kra91] M. A. Kramer. "Nonlinear principal component analysis using autoassociative neural networks". *AIChE Journal* **37**, 233–243 (1991) (cit. on p. 25).

[ROA17] J. Romero, J. P. Olson, and A. Aspuru-Guzik. "Quantum autoencoders for efficient compression of quantum data". *Quantum Science and Technology* **2**, 045001 (2017) (cit. on pp. 25, 27, 28, 36, 63).

[PTP19] A. Pepper, N. Tischler, and G. J. Pryde. "Experimental realization of a quantum autoencoder: The compression of qutrits via machine learning". *Physical Review Letters* **122**, 060501 (2019) (cit. on p. 26).

[Llo+20] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran. "Quantum embeddings for machine learning". *arXiv preprint arXiv:2001.03622* (2020) (cit. on p. 26).

[LC20] R. LaRose and B. Coyle. "Robust data encodings for quantum classifiers". *Physical Review A* **102**, 032420 (2020) (cit. on p. 26).

[SSM21] M. Schuld, R. Sweke, and J. J. Meyer. "Effect of data encoding on the expressive power of variational quantum-machine-learning models". *Physical Review A* **103**, 032430 (2021) (cit. on pp. 26, 63).

[GTN21] T. Goto, Q. H. Tran, and K. Nakajima. "Universal Approximation Property of Quantum Machine Learning Models in Quantum-Enhanced Feature Spaces". *Physical Review Letters* **127**, 090506 (2021) (cit. on pp. 26, 63).

[Hav+19] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta. "Supervised learning with quantum-enhanced feature spaces". *Nature* **567**, 209–212 (2019) (cit. on pp. 26, 63).

[PS+20a] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre. "Data re-uploading for a universal quantum classifier". *Quantum* **4**, 226 (2020) (cit. on pp. 26, 29, 63, 100).

[Eft+20] S. Efthymiou, S. Ramos-Calderer, C. Bravo-Prieto, A. Pérez-Salinas, D. García-Martín, A. Garcia-Saez, J. I. Latorre, and S. Carrazza. "Quantum-TII/qibo on Github" (2020) (cit. on p. 30).

[Eft+21b] S. Efthymiou, S. Ramos-Calderer, C. Bravo-Prieto, A. Pérez-Salinas, D. García-Martín, A. Garcia-Saez, J. I. Latorre, and S. Carrazza. "Qibo: a framework for quantum simulation with hardware acceleration". *Quantum Science and Technology* **7**, 015018 (2021) (cit. on pp. 30, 65).

[NW06] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science and Business Media. (2006) (cit. on p. 30).

[AD12] D. A. Abanin and E. Demler. "Measuring entanglement entropy of a generic many-body system with a quantum switch". *Physical Review Letters* **109**, 020504 (2012) (cit. on p. 35).

[Isl+15] R. Islam, R. Ma, P. M. Preiss, M. E. Tai, A. Lukin, M. Rispoli, and M. Greiner. "Measuring entanglement entropy in a quantum many-body system". *Nature* **528**, 77–83 (2015) (cit. on p. 35).

[EK95] A. Ekert and P. L. Knight. "Entangled quantum systems and the Schmidt decomposition". *American Journal of Physics* **63**, 415–423 (1995) (cit. on p. 35).

[Per06] A. Peres. *Quantum theory: concepts and methods*. Springer Science and Business Media. (2006) (cit. on p. 35).

[Reb+18] P. Rebentrost, A. Steffens, I. Marvian, and S. Lloyd. "Quantum singular-value decomposition of nonsparse low-rank matrices". *Physical Review A* **97**, 012327 (2018) (cit. on p. 35).

[LaR+18] R. LaRose, A. Tikku, É. O'Neel-Judy, L. Cincio, and P. J. Coles. "Variational quantum state diagonalization". *npj Quantum Information* **5**, 1–10 (2018) (cit. on pp. 35, 53, 54).

[JST17] S. Johri, D. S. Steiger, and M. Troyer. "Entanglement spectroscopy on a quantum computer". *Physical Review B* **96**, 195136 (2017) (cit. on p. 35).

[SCC19] Y. Subaşı, L. Cincio, and P. J. Coles. "Entanglement spectroscopy with a depth-two quantum circuit". *Journal of Physics A: Mathematical and Theoretical* **52**, 044001 (2019) (cit. on p. 35).

[DSW18] S. Das, G. Siopsis, and C. Weedbrook. "Continuous-variable quantum gaussian process regression and quantum singular value decomposition of nonsparse low-rank matrices". *Physical Review A* **97**, 022315 (2018) (cit. on p. 35).

[Car+20] J. Carolan, M. Mohseni, J. P. Olson, M. Prabhu, C. Chen, D. Bunandar, M. Y. Niu, N. C. Harris, F. N. Wong, M. Hochberg, et al. "Variational quantum unsampling on a quantum photonic processor". *Nature Physics* **16**, 322–327 (2020) (cit. on p. 36).

[PS+20b] A. Pérez-Salinas, D. García-Martín, C. Bravo-Prieto, and J. I. Latorre. "Measuring the tangle of three-qubit states". *Entropy* **22**, 436 (2020) (cit. on p. 36).

[Szo+22] T. Szołdra, P. Sierant, M. Lewenstein, and J. Zakrzewski. "Unsupervised detection of decoupled subspaces: many-body scars and beyond". *arXiv preprint arXiv:2201.07151* (2022) (cit. on p. 36).

[Rén61] A. Rényi. *On measures of entropy and information*. University of California Press. (1961) (cit. on p. 37).

[MRL08] M. Mohseni, A. T. Rezakhani, and D. A. Lidar. "Quantum-process tomography: Resource analysis of different strategies". *Physical Review A* **77**, 032322 (2008) (cit. on p. 37).

[WHT15] D. Wecker, M. B. Hastings, and M. Troyer. "Progress towards practical quantum variational algorithms". *Physical Review A* **92**, 042303 (2015) (cit. on p. 40).

[Mol+18] N. Moll, P. Barkoutsos, L. S. Bishop, J. M. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn, et al. "Quantum optimization using variational algorithms on near-term quantum devices". *Quantum Science and Technology* **3**, 030503 (2018) (cit. on p. 40).

[Ami+08] L. Amico, R. Fazio, A. Osterloh, and V. Vedral. "Entanglement in many-body systems". *Reviews of Modern Physics* **80**, 517 (2008) (cit. on p. 40).

[BŻ17] I. Bengtsson and K. Życzkowski. *Geometry of quantum states: an introduction to quantum entanglement*. Cambridge University Press. (2017) (cit. on p. 40).

[CLLG19] A. Cervera-Lierta, J. I. Latorre, and D. Goyeneche. "Quantum circuits for maximally entangled states". *Physical Review A* **100**, 022342 (2019) (cit. on p. 41).

[Cer+20a] M Cerezo, K. Sharma, A. Arrasmith, and P. J. Coles. “Variational Quantum State Eigensolver”. *arXiv preprint arXiv:2004.01372* (2020) (cit. on p. 43).

[WSW21] X. Wang, Z. Song, and Y. Wang. “Variational quantum singular value decomposition”. *Quantum* **5**, 483 (2021) (cit. on p. 43).

[Bis06] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer. (2006) (cit. on p. 45).

[Alp10] E. Alpaydin. *Introduction to Machine Learning*. The MIT Press. (2010) (cit. on p. 45).

[Eva10] L. C. Evans. *Partial differential equations*. American Mathematical Society. (2010) (cit. on p. 45).

[Bre95] O. Bretscher. *Linear Algebra With Applications*. Prentice Hall, Upper Saddle River NJ. (1995) (cit. on p. 45).

[SS11] D. A. Spielman and N. Srivastava. “Graph Sparsification by Effective Resistances”. *SIAM Journal on Computing* **40**, 1913–1926 (2011) (cit. on p. 45).

[Amb10] A. Ambainis. “Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations”. *arXiv preprint arXiv:1010.4458* (2010) (cit. on pp. 45, 58, 60).

[SSO19] Y. Subaşı, R. D. Somma, and D. Orsucci. “Quantum Algorithms for Systems of Linear Equations Inspired by Adiabatic Quantum Computing”. *Physical Review Letters* **122**, 060504 (2019) (cit. on pp. 45, 48, 50, 58, 60).

[CKS17] A. Childs, R. Kothari, and R. Somma. “Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision”. *SIAM Journal on Computing* **46**, 1920–1950 (2017) (cit. on pp. 45, 58, 60).

[CGJ18] S. Chakraborty, A. Gilyén, and S. Jeffery. “The power of block-encoded matrix powers: improved regression techniques via faster Hamiltonian simulation”. *arXiv preprint arXiv:1804.01973* (2018) (cit. on pp. 45, 58).

[WZP18] L. Wossnig, Z. Zhao, and A. Prakash. “Quantum Linear System Algorithm for Dense Matrices”. *Physical Review Letters* **120**, 050502 (2018) (cit. on p. 45).

[Zhe+17] Y. Zheng, C. Song, M.-C. Chen, B. Xia, W. Liu, et al. “Solving Systems of Linear Equations with a Superconducting Quantum Processor”. *Physical Review Letters* **118**, 210504 (2017) (cit. on p. 45).

[LJL19] Y. Lee, J. Joo, and S. Lee. “Hybrid quantum linear equation algorithm and its experimental test on IBM Quantum Experience”. *Scientific Reports* **9**, 4778 (2019) (cit. on p. 45).

[Pan+14] J. Pan, Y. Cao, X. Yao, Z. Li, C. Ju, et al. “Experimental realization of quantum algorithm for solving linear systems of equations”. *Physical Review A* **89**, 022313 (2014) (cit. on p. 45).

[Cai+13] X.-D. Cai, C. Weedbrook, Z.-E. Su, M.-C. Chen, M. Gu, et al. “Experimental Quantum Computing to Solve Systems of Linear Equations”. *Physical Review Letters* **110**, 230501 (2013) (cit. on p. 45).

[Bar+14] S. Barz, I. Kassal, M. Ringbauer, Y. O. Lipp, B. Dakić, et al. “A two-qubit photonic quantum processor and its application to solving systems of linear equations”. *Scientific Reports* **4**, 6115 (2014) (cit. on p. 45).

[Wen+19] J. Wen, X. Kong, S. Wei, B. Wang, T. Xin, and G. Long. “Experimental realization of quantum algorithms for a linear system inspired by adiabatic quantum computing”. *Physical Review A* **99**, 012320 (2019) (cit. on p. 45).

[Ans+19] E. Anschuetz, J. Olson, A. Aspuru-Guzik, and Y. Cao. “Variational quantum factoring”. *International Workshop on Quantum Technology and Optimization Problems*, 74–85 (2019) (cit. on p. 46).

[HWB19] O. Higgott, D. Wang, and S. Brierley. “Variational Quantum Computation of Excited States”. *Quantum* **3**, 156 (2019) (cit. on p. 46).

[Jon+19] T. Jones, S. Endo, S. McArdle, X. Yuan, and S. C. Benjamin. “Variational quantum algorithms for discovering Hamiltonian spectra”. *Physical Review A* **99**, 062304 (2019) (cit. on p. 46).

[Pra+21] P. Pravatto, D. Castaldo, F. Gallina, B. Fresch, S. Corni, and G. J. Moro. “Quantum computing for classical problems: variational quantum eigensolver for activated processes”. *New Journal of Physics* **23**, 123045 (2021) (cit. on p. 46).

[LB17] Y. Li and S. C. Benjamin. “Efficient Variational Quantum Simulator Incorporating Active Error Minimization”. *Physical Review X* **7**, 021050 (2017) (cit. on p. 46).

[Kok+19] C. Kokail, C. Maier, R. van Bijnen, T. Brydges, M. K. Joshi, P. Jurcevic, C. A. Muschik, P. Silvi, R. Blatt, C. F. Roos, and P. Zoller. “Self-verifying variational quantum simulation of lattice models”. *Nature* **569**, 355–360 (2019) (cit. on p. 46).

[Hey+19] K. Heya, K. M. Nakanishi, K. Mitarai, and K. Fujii. “Subspace variational quantum simulator”. *arXiv preprint arXiv:1904.08566* (2019) (cit. on p. 46).

[Yua+19] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin. “Theory of variational quantum simulation”. *Quantum* **3**, 191 (2019) (cit. on p. 46).

[Cî+20] C. Cîrstoiu, Z. Holmes, J. Iosue, L. Cincio, P. J. Coles, and A. Sornborger. “Variational fast forwarding for quantum simulation beyond the coherence time”. *npj Quantum Information* **6**, 82 (2020) (cit. on p. 46).

[BVC21] S. Barison, F. Vicentini, and G. Carleo. “An efficient quantum algorithm for the time evolution of parameterized circuits”. *Quantum* **5**, 512 (2021) (cit. on p. 46).

[Tan+22] X. Tang, C. Lyu, J. Li, X. Xu, M.-H. Yung, and A. Bayat. “Variational quantum simulation of long-range interacting systems”. *arXiv preprint arXiv:2203.14281* (2022) (cit. on p. 46).

[JB22] T. Jones and S. C. Benjamin. “Robust quantum compilation and circuit optimisation via energy minimisation”. *Quantum* **6**, 628 (2022) (cit. on p. 46).

[Arr+19] A. Arrasmith, L. Cincio, A. T. Sornborger, W. H. Zurek, and P. J. Coles. “Variational consistent histories as a hybrid algorithm for quantum foundations”. *Nature Communications* **10**, 3438 (2019) (cit. on p. 46).

[Kot+21] K. Kottmann, F. Metz, J. Fraxanet, and N. Baldelli. “Variational quantum anomaly detection: Unsupervised mapping of phase diagrams on a physical quantum computer”. *Physical Review Research* **3**, 043184 (2021) (cit. on p. 46).

[Joh+17] 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) (cit. on p. 46).

[LCM22] D. F. Locher, L. Cardarelli, and M. Müller. “Quantum Error Correction with Quantum Autoencoders”. *arXiv preprint arXiv:2202.00555* (2022) (cit. on p. 46).

[Cer+20b] M. Cerezo, A. Poremba, L. Cincio, and P. J. Coles. “Variational quantum fidelity estimation”. *Quantum* **4**, 248 (2020) (cit. on p. 46).

[Che+21] R. Chen, Z. Song, X. Zhao, and X. Wang. “Variational quantum algorithms for trace distance and fidelity estimation”. *Quantum Science and Technology* **7**, 015019 (2021) (cit. on p. 46).

[Koc+20] B. Koczor, S. Endo, T. Jones, Y. Matsuzaki, and S. C. Benjamin. “Variational-state quantum metrology”. *New Journal of Physics* **22**, 083038 (2020) (cit. on p. 46).

[GECP13] J. C. Garcia-Escartin and P. Chamorro-Posada. “Swap test and Hong-Ou-Mandel effect are equivalent”. *Physical Review A* **87**, 052330 (2013) (cit. on pp. 51, 52).

[Llo18] S. Lloyd. “Quantum approximate optimization is computationally universal”. *arXiv preprint arXiv:1812.11075* (2018) (cit. on p. 54).

[Wan+18] Z. Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel. “Quantum approximate optimization algorithm for MaxCut: A fermionic view”. *Physical Review A* **97**, 022304 (2018) (cit. on p. 54).

[Cro18] G. E. Crooks. “Performance of the quantum approximate optimization algorithm on the maximum cut problem”. *arXiv preprint arXiv:1811.08419* (2018) (cit. on p. 54).

[Zho+20b] 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”. *Physical Review X* **10**, 021067 (2020) (cit. on p. 54).

[HG17] Y. He and H. Guo. “The boundary effects of transverse field Ising model”. *Journal of Statistical Mechanics: Theory and Experiment* **2017**, 093101 (2017) (cit. on p. 55).

[Sha+20b] K. Sharma, S. Khatri, M. Cerezo, and P. Coles. “Noise resilience of variational quantum compiling”. *New Journal of Physics* **22**, 043006 (2020) (cit. on p. 60).

[Ber+07] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders. “Efficient quantum algorithms for simulating sparse Hamiltonians”. *Communications in Mathematical Physics* **270**, 359–371 (2007) (cit. on p. 60).

[AA17] Y. Atia and D. Aharonov. “Fast-forwarding of Hamiltonians and exponentially precise measurements”. *Nature Communications* **8**, 1572 (2017) (cit. on p. 60).

[Xu+21] X. Xu, J. Sun, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan. “Variational algorithms for linear algebra”. *Science Bulletin* **66**, 2181–2188 (2021) (cit. on p. 60).

[HBR21] H.-Y. Huang, K. Bharti, and P. Rebentrost. “Near-term quantum algorithms for linear systems of equations with regression loss functions”. *New Journal of Physics* **23**, 113021 (2021) (cit. on p. 60).

[PJ+21a] A. Pellow-Jarman, I. Sinayskiy, A. Pillay, and F. Petruccione. “A comparison of various classical optimizers for a variational quantum linear solver”. *Quantum Information Processing* **20**, 1–14 (2021) (cit. on p. 60).

[PJ+21b] A. Pellow-Jarman, I. Sinayskiy, A. Pillay, and F. Petruccione. “Near Term Algorithms for Linear Systems of Equations”. *arXiv preprint arXiv:2108.11362* (2021) (cit. on p. 60).

[PWK22] H. Patil, Y. Wang, and P. S. Krstić. "Variational quantum linear solver with a dynamic ansatz". *Physical Review A* **105**, 012423 (2022) (cit. on p. 60).

[Asf+19] A. Asfaw, L. Bello, Y. Ben-Haim, S. Bravyi, L. Capelluto, A. C. Vazquez, J. Ceroni, J. Gambetta, S. Garion, L. Gil, et al. *Learn Quantum Computation Using Qiskit*. (2019) (cit. on p. 60).

[Mar19] A. Mari. *PennyLane: Variational Quantum Linear Solver*. (2019) (cit. on p. 60).

[Aru+19] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, et al. "Quantum supremacy using a programmable superconducting processor". *Nature* **574**, 505–510 (2019) (cit. on p. 63).

[Zho+20a] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, et al. "Quantum computational advantage using photons". *Science* **370**, 1460–1463 (2020) (cit. on p. 63).

[Bia+17] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd. "Quantum machine learning". *Nature* **549**, 195–202 (2017) (cit. on p. 63).

[SP18] M. Schuld and F. Petruccione. *Supervised learning with quantum computers*. Springer. (2018) (cit. on p. 63).

[WBL12] N. Wiebe, D. Braun, and S. Lloyd. "Quantum algorithm for data fitting". *Physical Review Letters* **109**, 050505 (2012) (cit. on p. 63).

[LMR13] S. Lloyd, M. Mohseni, and P. Rebentrost. "Quantum algorithms for supervised and unsupervised machine learning". *arXiv preprint arXiv:1307.0411* (2013) (cit. on p. 63).

[RML14] P. Rebentrost, M. Mohseni, and S. Lloyd. "Quantum Support Vector Machine for Big Data Classification". *Physical Review Letters* **113**, 130503 (2014) (cit. on p. 63).

[KP20] I. Kerenidis and A. Prakash. "Quantum gradient descent for linear systems and least squares". *Physical Review A* **101**, 022316 (2020) (cit. on p. 63).

[Ben+19c] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini. "Parameterized quantum circuits as machine learning models". *Quantum Science and Technology* **4**, 043001 (2019) (cit. on p. 63).

[Lar+21] M. Larocca, N. Ju, D. García-Martín, P. J. Coles, and M Cerezo. "Theory of overparametrization in quantum neural networks". *arXiv preprint arXiv:2109.11676* (2021) (cit. on p. 63).

[PS+21a] A. Pérez-Salinas, D. López-Núñez, A. García-Sáez, P. Forn-Díaz, and J. I. Latorre. "One qubit as a universal approximant". *Physical Review A* **104**, 012405 (2021) (cit. on p. 63).

[Sch+20] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe. "Circuit-centric quantum classifiers". *Physical Review A* **101**, 032308 (2020) (cit. on p. 63).

[Dut+21] T. Dutta, A. Pérez-Salinas, J. P. S. Cheng, J. I. Latorre, and M. Mukherjee. "Single-qubit universal classifier implemented on an ion-trap quantum device". *arXiv preprint arXiv:2106.14059* (2021) (cit. on p. 63).

[BP21] C. Bravo-Prieto. "Quantum autoencoders with enhanced data encoding". *Machine Learning: Science and Technology* **2**, 035028 (2021) (cit. on p. 63).

[CW21] C. Cao and X. Wang. "Noise-Assisted Quantum Autoencoder". *Physical Review Applied* **15**, 054012 (2021) (cit. on p. 63).

[DT21] Y. Du and D. Tao. "On exploring practical potentials of quantum auto-encoder with advantages". *arXiv preprint arXiv:2106.15432* (2021) (cit. on p. 63).

[Ben+19a] M. Benedetti, D. Garcia-Pintos, O. Perdomo, V. Leyton-Ortega, Y. Nam, and A. Perdomo-Ortiz. "A generative modeling approach for benchmarking and training shallow quantum circuits". *npj Quantum Information* **5**, 1–9 (2019) (cit. on p. 63).

[HDP19] K. E. Hamilton, E. F. Dumitrescu, and R. C. Pooser. "Generative model benchmarks for superconducting qubits". *Physical Review A* **99**, 062323 (2019) (cit. on p. 63).

[Coy+20] B. Coyle, D. Mills, V. Danos, and E. Kashefi. "The Born supremacy: quantum advantage and training of an Ising Born machine". *npj Quantum Information* **6**, 1–11 (2020) (cit. on p. 63).

[Coy+21] B. Coyle, M. Henderson, J. C. J. Le, N. Kumar, M. Paini, and E. Kashefi. "Quantum versus classical generative modelling in finance". *Quantum Science and Technology* **6**, 024013 (2021) (cit. on p. 63).

[DDK18] P.-L. Dallaire-Demers and N. Killoran. "Quantum generative adversarial networks". *Physical Review A* **98**, 012324 (2018) (cit. on pp. 63, 65).

[LW18] S. Lloyd and C. Weedbrook. "Quantum generative adversarial learning". *Physical Review Letters* **121**, 040502 (2018) (cit. on p. 63).

[Goo+14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. "Generative adversarial nets". *Advances in Neural Information Processing Systems*, **17** (2014) (cit. on pp. 63, 64).

[ZLW19] C. Zoufal, A. Lucchi, and S. Woerner. "Quantum generative adversarial networks for learning and loading random distributions". *npj Quantum Information* **5**, 1–9 (2019) (cit. on p. 64).

[Zen+19] J. Zeng, Y. Wu, J.-G. Liu, L. Wang, and J. Hu. "Learning and inference on generative adversarial quantum circuits". *Physical Review A* **99**, 052306 (2019) (cit. on p. 64).

[Hu+19] 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". *Science advances* **5**, eaav2761 (2019) (cit. on pp. 64, 65).

[Ben+19b] M. Benedetti, E. Grant, L. Wossnig, and S. Severini. "Adversarial quantum circuit learning for pure state approximation". *New Journal of Physics* **21**, 043023 (2019) (cit. on pp. 64, 65).

[Sit+20] H. Situ, Z. He, Y. Wang, L. Li, and S. Zheng. "Quantum generative adversarial network for generating discrete distribution". *Information Sciences* **538**, 193–208 (2020) (cit. on p. 64).

[RAG21] J. Romero and A. Aspuru-Guzik. "Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions". *Advanced Quantum Technologies* **4**, 2000003 (2021) (cit. on pp. 64, 65).

[Niu+21] M. Y. Niu, A. Zlokapa, M. Broughton, S. Boixo, M. Mohseni, V. Smelyanskyi, and H. Neven. "Entangling Quantum Generative Adversarial Networks". *arXiv preprint arXiv:2105.00080* (2021) (cit. on pp. 64, 65).

[KLA21] T. Karras, S. Laine, and T. Aila. "A Style-Based Generator Architecture for Generative Adversarial Networks". *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43**, 4217–4228 (2021) (cit. on p. 64).

[PS+21b] A. Pérez-Salinas, J. Cruz-Martinez, A. A. Alhajri, and S. Carrazza. "Determining the proton content with a quantum computer". *Physical Review D* **103**, 034027 (2021) (cit. on p. 64).

[Gua+21] W. Guan, G. Perdue, A. Pesah, M. Schuld, K. Terashi, S. Vallecorsa, and J.-R. Vlimant. "Quantum machine learning in high energy physics". *Machine Learning: Science and Technology* **2**, 011003 (2021) (cit. on p. 64).

[Cha+21a] S. Y. Chang, S. Herbert, S. Vallecorsa, E. F. Combarro, and R. Duncan. "Dual-Parameterized Quantum Circuit GAN Model in High Energy Physics". *EPJ Web of Conferences* **251**, 03050 (2021) (cit. on p. 64).

[Cha+21b] S. Y. Chang, S. Vallecorsa, E. F. Combarro, and F. Carminati. "Quantum Generative Adversarial Networks in a Continuous-Variable Architecture to Simulate High Energy Physics Detectors". *arXiv preprint arXiv:2101.11132* (2021) (cit. on p. 64).

[Bel+21] V. Belis, S. González-Castillo, C. Reissel, S. Vallecorsa, E. F. Combarro, G. Dissertori, and F. Reiter. "Higgs analysis with quantum classifiers". *EPJ Web of Conferences* **251**, 03070 (2021) (cit. on p. 64).

[Agl+22] G. Agliardi, M. Grossi, M. Pellen, and E. Prati. "Quantum integration of elementary particle processes". *arXiv preprint arXiv:2201.01547* (2022) (cit. on p. 64).

[AS22] J. Y. Araz and M. Spannowsky. "Classical versus Quantum: comparing Tensor Network-based Quantum Circuits on LHC data". *arXiv preprint arXiv:2202.10471* (2022) (cit. on p. 64).

[DH22] A. Delgado and K. E. Hamilton. "Unsupervised Quantum Circuit Learning in High Energy Physics". *arXiv preprint arXiv:2203.03578* (2022) (cit. on p. 64).

[Del+22] A. Delgado, K. E. Hamilton, J.-R. Vlimant, D. Magano, Y. Omar, P. Bargassa, A. Francis, A. Gianelle, L. Sestini, D. Lucchesi, et al. "Quantum Computing for Data Analysis in High-Energy Physics". *arXiv preprint arXiv:2203.08805* (2022) (cit. on p. 64).

[BPW19] A. Butter, T. Plehn, and R. Winterhalder. "How to GAN LHC events". *SciPost Physics* **7**, 075 (2019) (cit. on p. 64).

[BPW20] A. Butter, T. Plehn, and R. Winterhalder. "How to GAN event subtraction". *SciPost Physics Core* **3**, 009 (2020) (cit. on p. 64).

[Bel+20] M. Bellagente, A. Butter, G. Kasieczka, T. Plehn, and R. Winterhalder. "How to GAN away detector effects". *SciPost Physics* **8**, 070 (2020) (cit. on p. 64).

[BP20] A. Butter and T. Plehn. "Generative Networks for LHC events". *arXiv preprint arXiv:2008.08558* (2020) (cit. on p. 64).

[Bal+21] P. Baldi, L. Blecher, A. Butter, J. Collado, J. N. Howard, F. Keilbach, T. Plehn, G. Kasieczka, and D. Whiteson. "How to GAN Higher Jet Resolution". *arXiv preprint arXiv:2012.11944* (2021) (cit. on p. 64).

[Bac+21] M. Backes, A. Butter, T. Plehn, and R. Winterhalder. "How to GAN Event Unweighting". *SciPost Physics* **10**, 089 (2021) (cit. on p. 64).

[But+21] A. Butter, S. Diefenbacher, G. Kasieczka, B. Nachman, and T. Plehn. "GANplifying event samples". *SciPost Physics* **10**, 139 (2021) (cit. on p. 64).

[Eft+21a] S. Efthymiou, S. Carrazza, S. Ramos, bpcarlos, AdrianPerezSalinas, D. García-Martín, Paul, J. Serrano, and atomicprinter. *qiboteam/qibo: Qibo 0.1.6-rc1*. (2021) (cit. on p. 65).

[Aba+15] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. (2015) (cit. on p. 66).

[hep+21] afrancis heplat, C. Bravo-Prieto, S. Carrazza, M. Cè, J. Baglio, and d-m grabowska. *QTI-TH/style-qgan: v1.0.0*. (2021) (cit. on pp. 66, 67).

[KL51] S. Kullback and R. A. Leibler. "On information and sufficiency". *The Annals of Mathematical Statistics* **22**, 79–86 (1951) (cit. on p. 68).

[FA+18] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan. "Synthetic data augmentation using GAN for improved liver lesion classification". *IEEE 15th International Symposium on Biomedical Imaging*, 289–293 (2018) (cit. on p. 69).

[TA19] F. H. K. d. S. Tanaka and C. Aranha. "Data Augmentation Using GANs". *arXiv preprint arXiv:1904.09135* (2019) (cit. on p. 69).

[Alw+14] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H. S. Shao, T. Stelzer, P. Torrielli, and M. Zaro. "The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations". *Journal of High Energy Physics* **07**, 079 (2014) (cit. on p. 72).

[Fre+18] R. Frederix, S. Frixione, V. Hirschi, D. Pagani, H. S. Shao, and M. Zaro. "The automation of next-to-leading order electroweak calculations". *Journal of High Energy Physics* **07**, 185 (2018) (cit. on p. 72).

[YJ00] I.-K. Yeo and R. A. Johnson. "A new family of power transformations to improve normality or symmetry". *Biometrika* **87**, 954–959 (2000) (cit. on p. 73).

[Ped+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, Dubourg, et al. "Scikit-learn: Machine Learning in Python". *Journal of Machine Learning Research* **12**, 2825–2830 (2011) (cit. on p. 73).

[Ale+19] G. Aleksandrowicz, T. Alexander, P. Barkoutsos, L. Bello, Y. Ben-Haim, D. Bucher, F. J. Cabrera-Hernández, J. Carballo-Franquis, A. Chen, C.-F. Chen, et al. *Qiskit: An Open-source Framework for Quantum Computing*. (2019) (cit. on p. 75).

[RC+21b] S. Ramos-Calderer, A. Pérez-Salinas, D. García-Martín, C. Bravo-Prieto, J. Cortada, J. Planaguma, and J. I. Latorre. "Quantum unary approach to option pricing". *Physical Review A* **103**, 032414 (2021) (cit. on p. 100).

[RC+21a] S. Ramos-Calderer, E. Bellini, J. I. Latorre, M. Manzano, and V. Mateu. "Quantum search for scaled hash function preimages". *Quantum Information Processing* **20**, 1–28 (2021) (cit. on p. 100).

[Nie+06] M. A. Nielsen, M. R. Dowling, M. Gu, and A. C. Doherty. "Quantum Computation as Geometry". *Science* **311**, 1133–1135 (2006) (cit. on p. 101).

[DN08] M. R. Dowling and M. A. Nielsen. "The Geometry of Quantum Computation". *Quantum Information and Computation* **8**, 861–899 (2008) (cit. on p. 101).

[Pow78] M. J. D. Powell. "A fast algorithm for nonlinearly constrained optimization calculations". *Numerical analysis*. Springer. (1978) (cit. on p. 107).

Part V

APPENDIX

A

Qibo: a framework for quantum simulation

The quantum computer simulation on classical hardware is still quite relevant in the current research stage. Thanks to simulation, researchers can prototype and study a priori the behavior of new algorithms on quantum hardware. In this appendix, we present `Qibo`, a framework for quantum simulation. Specifically, `Qibo` is an open-source software for fast evaluation of quantum circuits and adiabatic evolution, which takes full advantage of hardware accelerators.

The `Qibo` framework is designed to delegate all complicated aspects of hardware or platform implementation to the library so researchers can focus on the problem and quantum algorithms at hand. This software is designed from scratch with simulation performance, code simplicity, and a user-friendly interface as target goals. The high-level API receives instructions from the user and automatically allocates and executes the code on specific, optimized backends. The API can receive simple instructions, *e.g.* quantum gates or circuits, and also more sophisticated models, such as a variational quantum algorithm. In the following, a short review of the `Qibo` basics for gate-based applications is presented.

CIRCUITS

- Initialization. `Qibo` simulates the behavior of quantum circuits using dense complex state vectors $\psi(\sigma_1, \sigma_2, \ldots, \sigma_N) \in \mathbb{C}^{2^N}$ in the computational basis where $\sigma_i \in \{0, 1\}$ and N is the total number of qubits in the circuit. The initialization of a quantum circuit is the following:

```python
from qibo.models import Circuit

# create a circuit for N=2 qubits
qubits = 2
circuit = Circuit(qubits)
state = circuit().numpy()

print(state)
>>> array([1.+0.j, 0.+0.j, 0.+0.j, 0.+0.j])
```

- GATES. Quantum gates are added to the circuit by acting on the state vector via matrix multiplication. Gates can be defined as acting on one or several qubits.

```
import numpy as np
from qibo import gates

qubits = 2
circuit = Circuit(qubits)

# add some gates in the circuit
circuit.add([gates.H(0), gates.X(1)])
circuit.add(gates.RX(0, theta=np.pi/6))
circuit.add(gates.CNOT(0, 1))

print(circuit.draw())
>>> q0: -H-RX-o-
>>> q1: -X----X-
```

- MEASUREMENTS. The measurement step is used to retrieve information from the quantum computation. Measurements can be allocated at any part of the circuit.

```
qubits = 2
circuit = Circuit(qubits)

circuit.add(gates.X(0))
circuit.add(gates.H(1))
circuit.add(gates.CNOT(1, 0))
circuit.add(gates.M(0)) # measurement in qubit 0
circuit.add(gates.H(1))
circuit.add(gates.M(1)) # measurement in qubit 1

print(circuit.draw())
>>> q0: -X-X-M---
>>> q1: -H-o-H-M-
```

- CALLBACKS. The callback functions allow the user to perform calculations on intermediate state vectors during a circuit execution. A callback example that is implemented in `Qibo` is entanglement entropy. This allows the user to track how entanglement changes as the state is propagated through the circuit's gates. Other callbacks implemented in `Qibo` include the `callbacks.Energy` which calculates the energy (expectation value of a Hamiltonian) of a state.

```
from qibo import callbacks

qubits = 2
circuit = Circuit(qubits)

# create entropy callback where qubit 0 is one subsystem
entropy = callbacks.EntanglementEntropy([0])

circuit.add(gates.CallbackGate(entropy)) # entropy calculation |00> state
circuit.add(gates.H(0))
circuit.add(gates.CallbackGate(entropy)) # entropy calculation |+> state
circuit.add(gates.CNOT(0, 1))
circuit.add(gates.CallbackGate(entropy)) # entropy calculation Bell state

final_state = circuit()
print(entropy[:])
>>> [0.0 0.0 1.0]
```

HAMILTONIANS

The Hamiltonian of a system is an operator corresponding to the total energy of that system. As we have seen in different chapters, Hamiltonians play a relevant role in variational quantum algorithms since cost functions can be defined as expectation values of such operators. Qibo users have to define Hamiltonians based on `qibo.core.hamiltonians.Hamiltonian` which uses the full matrix representation of the corresponding operator or `qibo.core.hamiltonians.SymbolicHamiltonian` which uses a more efficient term representation. Qibo provides pre-coded Hamiltonians for some common models, such as the transverse field Ising model and the Heisenberg model. In order to explore other problems, the user needs to define the Hamiltonian objects from scratch.

```
from qibo.hamiltonians import Hamiltonian
from qibo import matrices

# ZZ terms
matrix = np.kron(np.kron(matrices.Z, matrices.Z), np.kron(matrices.I,
↪   matrices.I))
matrix += np.kron(np.kron(matrices.I, matrices.Z), np.kron(matrices.Z,
↪   matrices.I))
matrix += np.kron(np.kron(matrices.I, matrices.I), np.kron(matrices.Z,
↪   matrices.Z))
matrix += np.kron(np.kron(matrices.Z, matrices.I), np.kron(matrices.I,
↪   matrices.Z))
# X terms
matrix += np.kron(np.kron(matrices.X, matrices.I), np.kron(matrices.I,
↪   matrices.I))
matrix += np.kron(np.kron(matrices.I, matrices.X), np.kron(matrices.I,
↪   matrices.I))
matrix += np.kron(np.kron(matrices.I, matrices.I), np.kron(matrices.X,
↪   matrices.I))
matrix += np.kron(np.kron(matrices.I, matrices.I), np.kron(matrices.I,
↪   matrices.X))

# Create Hamiltonian object
ham = Hamiltonian(4, matrix)
```

```
from qibo.hamiltonians import SymbolicHamiltonian
from qibo.symbols import X, Z

# define Hamiltonian using Qibo symbols
# ZZ terms
symbolic_ham = sum(Z(i) * Z(i + 1) for i in range(3))
# periodic boundary condition term
symbolic_ham += Z(0) * Z(3)
# X terms
symbolic_ham += sum(X(i) for i in range(4))

# define a Hamiltonian using the above form
ham = SymbolicHamiltonian(symbolic_ham)
# this Hamiltonian is memory efficient as it does not construct the
# full matrix

# the corresponding dense Hamiltonian which contains the full matrix can
# be constructed easily as
dense_ham = ham.dense
# and the matrix is accessed as ``dense_ham.matrix`` or ``ham.matrix``.
```

MODELS

`Qibo` provides several application-specific models, which are pre-defined architectures to implement seminal algorithms such as the variational quantum eigensolver [Per+14] or the quantum approximate optimization algorithm [FGG14].

[Per+14] Peruzzo et al., “A variational eigenvalue solver on a photonic quantum processor”

[FGG14] Farhi et al., “A quantum approximate optimization algorithm”

```
from qibo.hamiltonians import XXZ
from qibo.models import VQE

qubits = 2
circuit = Circuit(qubits)
circuit.add(gates.RY(0, theta=0))
circuit.add(gates.RY(1, theta=0))

# define XXZ Hamiltonian
hamiltonian = XXZ(qubits)

# define VQE model
vqe = VQE(circuit, hamiltonian)

initial_parameters = np.random.uniform(0, 2*np.pi, 2)
result = vqe.minimize(initial_parameters)
print('Found energy:', result[0])
>>> Found energy: -1.9999976950940028
```

ADVANCED EXAMPLES AND APPLICATIONS

`Qibo` aims to be a community-driven open project. As a starting point, several advanced examples and implementations were created to trigger contributions from users around the globe. Let us mention below a few of them:

- Scaling of variational quantum circuit depth for condensed matter systems explored in Chapter 2.
- Quantum autoencoder for data compression and Quantum autoencoders with enhanced data encoding, both explored in Chapter 3.
- Quantum singular value decomposer explored in Chapter 4.
- Class that implements the style-based quantum generative adversarial model explored in Chapter 6.
- Data reuploading for a universal quantum classifier [PS+20a].
- Quantum unary approach to option pricing [RC+21b].
- Shor’s factorization algorithm [Sho99].
- Grover’s Algorithm for solving a Toy Sponge Hash function [RC+21a].
- Adiabatic evolution for solving an Exact Cover problem.

[PS+20a] Pérez-Salinas et al., “Data reuploading for a universal quantum classifier”

[RC+21b] Ramos-Calderer et al., “Quantum unary approach to option pricing”

[Sho99] Shor, “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”

B

Solovay-Kitaev theorem

One of the significant challenges of quantum computation is to implement quantum algorithms efficiently. Mathematically, the complexity of an algorithm can be defined in the context of differential geometry relating distances in the manifold of unitary operations with the circuit complexity [Nie+06; DN08]. In quantum computing, we are limited to a specific set of quantum gates to perform arbitrary unitary operations. Thus, we have to find optimal combinations of these gates that approximate the desired operation. Approximating a unitary operation U, given a set of gates $\mathcal{G}$, means that we have to find $g_1 \cdots g_l \in \mathcal{G}$ such that $\|U - g_1 \cdots g_l\|$ is sufficiently small ($\|\cdot\|$ denotes a distance in the manifold of unitary operations such as the operator norm or the trace norm).

[Nie+06] Nielsen et al., "Quantum Computation as Geometry"

[DN08] Dowling and Nielsen, "The Geometry of Quantum Computation"

GROUND STATE ENERGY ERROR

In the following, we are going to relate the error ε between an approximated quantum circuit $\tilde{U}$ and the exact one U that produces a ground state $|\psi_0\rangle$ of some Hamiltonian H, with the error of the ground state energy. The error of the ground state energy may be defined as $|\tilde{E}_0 - E_0|$, where E_0 is the lowest eigenvalue of H, and $\tilde{E}_0$ is the expectation value $\langle\tilde{\psi}_0| H |\tilde{\psi}_0\rangle$, being $|\tilde{\psi}_0\rangle$ the approximated state given by $\tilde{U}$. We may assume as well that exists an ideal circuit U that maps our initial state to the exact ground state of a given Hamiltonian, although implementing U may not be efficient in terms of the number of qubits n.

Lemma 1. *Given a universal set of quantum gates $\mathcal{G}$ closed under inversion, a Hamiltonian H, and error $\varepsilon > 0$ it is possible to find a quantum circuit $\tilde{U}$ such that it can simulate an approximation for the ground state $|\tilde{\psi}_0\rangle$, with an error of the energy of $\mathcal{O}(\varepsilon^2)$ in a gate complexity of*

$$\mathcal{O}(\log^c(1/\varepsilon)), \tag{B.1}$$

for some constant c, $c \leq 4$

Proof. To prove this result we use the Solovay-Kitaev theorem and standard perturbation theory. Suppose that exists a quantum circuit U such that

$$U |0\rangle^{\otimes n} = |\psi_0\rangle . \tag{B.2}$$

Then, from the Solovay-Kitaev theorem it is possible to find an ε-approximation $\tilde{U}$ for U using $\mathcal{O}(log^c(1/\varepsilon))$ gates from our set $\mathcal{G}$. The approximated $\tilde{U}$ can be expressed as $\tilde{U} = e^{-i\varepsilon A}U$ for some bounded Hermitian matrix A ($||A|| < 1$). Expanding $\tilde{U}$ with the usual definition of the matrix exponentiation to the first order on ε, we compute the approximated state $|\tilde{\psi_0}\rangle$ of the exact groundstate

$$|\tilde{\psi_0}\rangle = \tilde{U}\,|0\rangle^{\otimes n} = |\psi_0\rangle - i\varepsilon A|\psi_0\rangle + \mathcal{O}(\varepsilon^2). \tag{B.3}$$

Recall that since A is bounded, $|\tilde{\psi_0}\rangle$ is ε-close to $|\psi_0\rangle$. Finally, it suffices to compute the energy of the state $|\tilde{\psi_0}\rangle$ as $\tilde{E_0} = \langle\tilde{\psi_0}|H\,|\tilde{\psi_0}\rangle$. Given that $E_0 = \langle\psi_0|H\,|\psi_0\rangle$, then

$$\tilde{E_0} = E_0 + \varepsilon^2 \langle\psi_0|AHA|\psi_0\rangle\ . \tag{B.4}$$

The terms $\mathcal{O}(\varepsilon)$ have canceled due to the hermicity of A and the change of sign produced by the conjugation of the imaginary unit i. Thus, the result $|\tilde{E_0} - E_0| = \mathcal{O}(\varepsilon^2)$ follows. □

VON NEUMANN ENTROPY ERROR

This result can be extended also to the Von Neumann entropy. Recall the definition of the Von Neumann entropy. Let $\mathcal{H}$ be a bipartite Hilbert space for two subsystems A and B, i.e $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$, then, ρ_0^A the reduced density matrix of a state $|\psi_0\rangle$ reads

$$\rho_0^A = Tr_B\,|\psi_0\rangle\,\langle\psi_0|\ . \tag{B.5}$$

The Von Neumann entropy of the bipartition can be computed as

$$S_0 = -Tr\rho_0^A \log_2 \rho_0^A = -\sum_i^{\chi} \lambda_i \log_2 \lambda_i\,, \tag{B.6}$$

where λ_i are the eigenvalues of ρ_0^A, and χ is the Schmidt rank.

The following result will extend the relationship between the error ε of the approximated quantum circuit $\tilde{U}$ with the entropy error $|\tilde{S}_0 - S_0|$, where S_0 corresponds to the entropy of the exact ground state and $\tilde{S}_0$ to the approximated one.

Lemma 2. *Given a universal set of quantum gates $\mathcal{G}$ closed under inversion, a Hamiltonian H, and error $\varepsilon > 0$ it is possible to find a quantum circuit $\tilde{U}$ such that it can simulate an approximation for the ground state $|\tilde{\psi_0}\rangle$, with an error of the Von Neumann entropy of $\mathcal{O}(\varepsilon)$ in a gate complexity of*

$$\mathcal{O}(\log^c(1/\varepsilon))\,, \tag{B.7}$$

for some constant c, $c \leq 4$.

Proof. Using the same construction as in Lemma 1, we may find an ε-approximation $\tilde{U}$ of the ideal circuit, that produces an approximated state $|\tilde{\psi_0}\rangle$. In order to compute $\tilde{S_0}$ it is useful to compute the density matrix of $|\tilde{\psi_0}\rangle$,

$$\tilde{\rho}_0^A = \rho_0^A + i\varepsilon Tr_B(-A|\psi_0\rangle\,\langle\psi_0| + |\psi_0\rangle\,\langle\psi_0|A) + \mathcal{O}(\varepsilon^2)\,. \tag{B.8}$$

Note that the terms of $\mathcal{O}(\varepsilon)$ does not cancel, thus $\|\tilde{\rho}_0^A - \rho_0^A\| = \mathcal{O}(\varepsilon)$. Let $\lambda_0, ..., \lambda_m$ and $\tilde{\lambda}_0, ..., \tilde{\lambda}_m$ be the eigenvalues of ρ_0^A and $\tilde{\rho}_0^A$, respectively. Then, the eigenvalues can be related as

$$\tilde{\lambda}_i = \lambda_i + \varepsilon c_i + \mathcal{O}(\varepsilon^2)\,, \tag{B.9}$$

where c_i is some constant such that $|\tilde{\lambda}_i - \lambda_i| = \mathcal{O}(\varepsilon)$, and the terms of higher order on ε are ignored. Finally, it suffices to compute the terms $\lambda_i \log_2 \lambda_i$. Expressing $\log_2 \tilde{\lambda}_i = \log_2 (\lambda_i(1 + \varepsilon c_i/\lambda_i))$, and using the Taylor expansion for the logarithm, we obtain that

$$\tilde{\lambda}_i \log_2 \tilde{\lambda}_i = \lambda_i \log_2 \lambda_i + c_i \varepsilon \log_2 \lambda_i + c_i \varepsilon + \mathcal{O}(\varepsilon^2)\,. \tag{B.10}$$

Then, summing over all the terms $\tilde{\lambda}_i \log_2 \tilde{\lambda}_i$, the result $|\tilde{S}_0 - S_0| = \mathcal{O}(\varepsilon)$ follows. ☐

Hence, we may conclude that if some unitary U accepts a polylogarithmic approximation $\tilde{U}$ up to some error $\mathcal{O}(\varepsilon)$, then we can approximate as well the ground state energy and the Von Neumann entropy up to $\mathcal{O}(\varepsilon^2)$ and $\mathcal{O}(\varepsilon)$, respectively.

C

Small scale implementations of the linear system problem

SCALING HEURISTICS FOR 8×8 SYSTEMS

Here we study the scaling of the variational quantum linear solver (VQLS) algorithm of Chapter 5 for 8×8 systems. For this purpose, we employed the ansatz of Figure C.1 (with randomly initialized parameters), and we numerically implement VQLS to solve the three different QLSPs with different degeneracy g in the minimum eigenvalue of A defined by:

$$A_1 = \frac{1}{8\kappa}[4(\kappa+1)\mathbb{1} + (\kappa-1)(Z_3 + Z_2 + 2Z_1)] \tag{C.1}$$

$$A_2 = \frac{1}{4\kappa}[2(\kappa+1)\mathbb{1} + (\kappa-1)(Z_3 + Z_2)] \tag{C.2}$$

$$A_3 = \frac{1}{2\kappa}[(\kappa+1)\mathbb{1} + (\kappa-1)Z_3]\,. \tag{C.3}$$

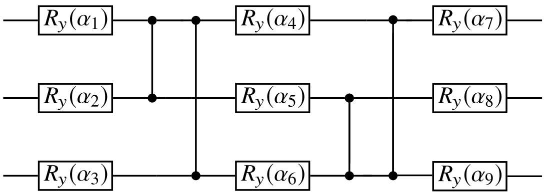

FIGURE C.1: Hardware-Efficient Ansatz used to solve the QLSPs in Eqs. C.1–C.4. Since A and $|b\rangle$ are real, $V(\vec{\alpha})$ contains only rotation around the y-axis $R_y(\alpha_i) = e^{-i\alpha_i Y/2}$, and control-$Z$ gates.

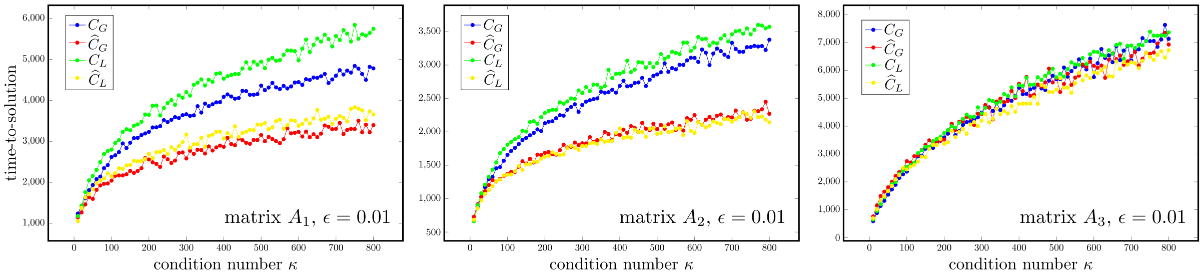

FIGURE C.2: Time-to-solution versus condition number κ. The time-to-solution is the mean number of executions needed to guarantee a desired precision ϵ. The QLSP is determined by $|b\rangle$ of C.4, and A given by: *Left:* Matrix A_1 of C.1. *Center:* Matrix A_2 of C.2. *Right:* Matrix A_3 of C.3. For each data point we ran and averaged 1000 instances of the VQLS algorithm. In all cases we trained the gate sequence by minimizing $\widehat{C}_G$, C_G, $\widehat{C}_L$, and C_L. As can be seen, the scaling in terms of the condition number κ appears to be efficient for all A matrices and for all cost functions.

The degeneracy of each matrix is $g = 1, 2, 4$, respectively. We remark that we considered different values of g to analyze if this parameter affects the VQLS performance. The state $|b\rangle$ is

$$|b\rangle = H^{\otimes 3} |000\rangle \ . \tag{C.4}$$

Figure C.2 shows results plotting time-to-solution versus κ for the aforementioned A matrices by training $\widehat{C}_G$, C_G, $\widehat{C}_L$, and C_L. In all cases we employed the operational meanings of our cost functions in Eqs. 5.10 and 5.11 of the main text for our certification procedure, i.e., to upper-bound the quantity ϵ. For each data point in Figure C.2, we implemented and averaged over 1000 runs of VQLS.

These results show that the κ scaling is efficient for the problems considered (regardless of the value of g considered). This is in agreement with the scaling observed in Chapter 5. It is worth noting that this efficient scaling holds for all of our cost functions. The unnormalized cost functions have slightly better performance for the A_1 and A_2 matrices, although all four cost functions perform similarly for A_3, indicating that the performances of different cost functions are problem-dependent.

IMPLEMENTATION WITH QAOA ANSATZ

We numerically analyze the VQLS scaling with κ when employing the QAOA ansatz. Since poorly conditioned matrices (i.e., large κ) are more difficult to invert, we expect that for fixed ϵ, the number of layers p must increase with κ. While this is generally true, we can also alleviate this issue by evolving with the driver Hamiltonian H_D for a longer time. This corresponds to scaling the parameters α_i for odd i in Eq. 5.27 of the main text by some value that grows with κ. As shown in Figure C.3(a) and (b), this scaling can indeed transform the cost landscape such that it contains more regions of low cost and thus makes optimization more likely to be successful.

In Figure C.3(c), we show the time-to-solution versus the condition number. Here, we consider the QLSP on three qubits defined by the A_2 matrix of C.2 and with $|b\rangle$ given by C.4. For this small scale-implementation we obtained the time-to-solution be exactly computing ϵ. The condition number was varied from $\kappa = 10^0$ to $\kappa = 10^3$. For each κ, VQLS was implemented 100 times with the parameters randomly initialized. For each of the three values of ϵ considered, the scaling with κ is sub-exponential. Hence, these results indicate that VQLS with QAOA also scales efficiently in the condition number κ. Finally, we emphasize that these results were obtained with only $p = 1$ round of QAOA and remark that additional rounds $p > 1$ may lead to better performance.

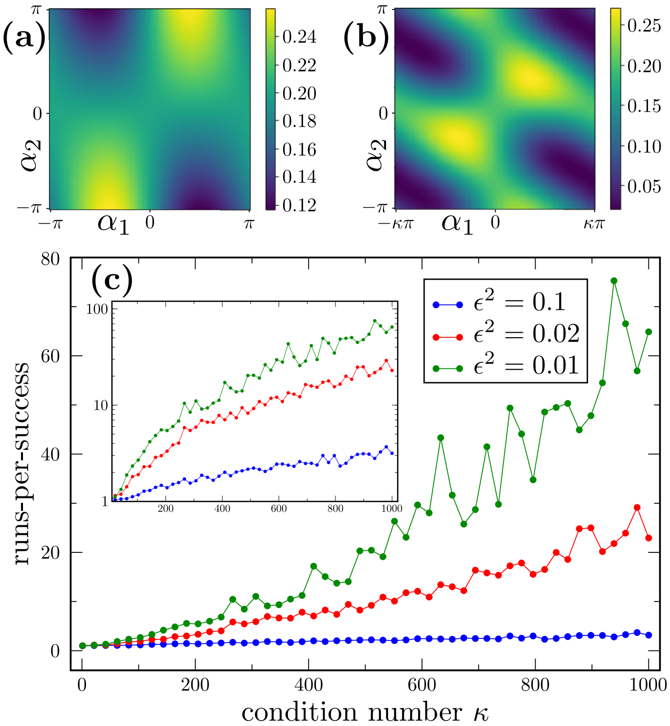

FIGURE C.3: (a) Landscape for $\widehat{C}_G$ with a QAOA ansatz of $p = 1$ layer and unscaled parameters α_1 and α_2. Here, α_1 (α_2) corresponds the the parameter in the driver (mixer) Hamiltonian. (b) Landscape for $\widehat{C}_G$ with a QAOA ansatz of $p = 1$ layer where α_1 was scaled by the condition number κ. In both cases the QLSP is defined by a randomly generated 4×4 matrix with condition number $\kappa \approx 11$, and with $|b\rangle$ given by C.4. The scaled landscape contains more regions of low cost and thus makes optimization more likely to be successful. (c) Time-to-solution versus condition number κ for the QLSP on three qubits defined by the A_2 matrix of C.2 and with $|b\rangle$ given by C.4. Three curves are shown for $\epsilon^2 = 0.10, 0.02$, and 0.01. The inset depicts the same data on a logarithmic scale. As can be seen from the inset, the scaling in κ is sub-exponential for each ϵ considered.

IMPLEMENTATIONS ON RIGETTI'S QUANTUM COMPUTER

Here we present additional implementations performed on Rigetti's quantum device *16Q Aspen-4*. We have considered different problem sizes, from 2×2 up to 32×32. We additionally recall that the matrices A and states $|b\rangle$ in these QLSP are such that the ansatz and the cost computing circuits are simplified.

First we present the results of a 32×32 (i.e., 5-qubit) implementation of VQLS using Rigetti's quantum chip *16Q Aspen-4*. We considered the QLSP defined by

$$A = \mathbb{1} + 0.2 X_1 Z_2 + 0.2 X_1 \,, \tag{C.5}$$

and $|b\rangle = H_1 H_3 H_4 H_5 |0\rangle^{\otimes 5}$. This particular choice of A and $|b\rangle$ is motivated from the fact that they lead to simplified ansatz and cost evaluation circuits. In particular, the ansatz considered consists of $R_y(\alpha_i)$ gates acting on each qubit.

The results are shown in Figure C.4. At each run of the VQLS algorithm, the parameters were initialized to random angles, and the classical optimization was performed with the Powell method [Pow78]. For every run, the local cost function C_L of Eq. 5.6 achieved a value of $\sim 7 \times 10^{-2}$ (hardware noise prevented further cost reduction). While this cost value led

[Pow78] Powell, "A fast algorithm for nonlinearly constrained optimization calculations"

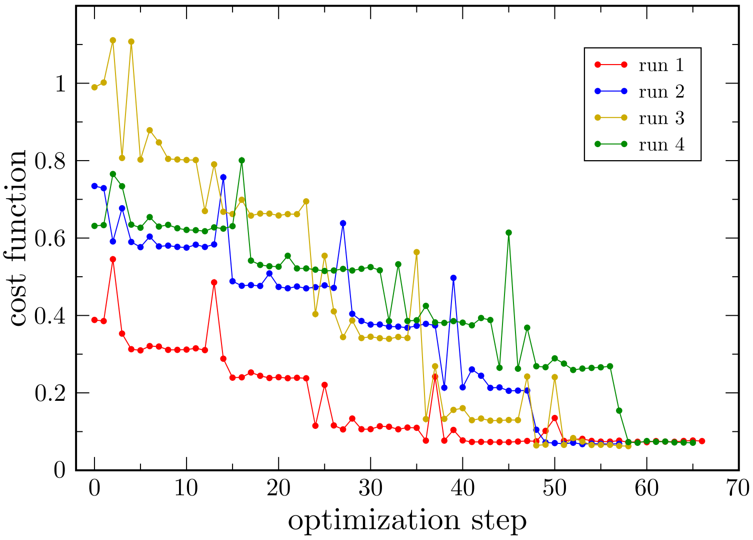

FIGURE C.4: Implementation of VQLS on Rigetti's quantum hardware. Cost function C_L is plotted versus number of optimization steps, where A is given by C.5. One can observe that for every run the cost function is reduced to a value of $\sim 7 \times 10^{-2}$. Due to noise present in the quantum device the cost does not go to zero.

to a trivial bound on ϵ via Eq. 5.10, we nevertheless found the solution $|x\rangle$ to be of high quality. We verified this by measuring the expectation value of different Hermitian observables M on the state $|x\rangle$ prepared on the quantum computer. According to Eq. 5.12, we can use $D(M)^2$ as a figure of merit to quantify the quality of our solution. For all M we considered, $D(M)^2$ was no larger than 0.01, and hence the results have a good agreement with the exact solution. See Table C.7 for all values of $D(M)^2$.

Figure C.5 shows the value of the cost function versus the number of optimization steps for different linear systems and several runs. It is worth mentioning that the cost function is reduced to values $\lesssim 0.1$ for every example, except for the case depicted in panel (b). In this particular case, the solution of the 2×2 linear system is $|x_0\rangle = |1\rangle$. Therefore, one may note the effect of relaxation to the state $|0\rangle$ in the quantum device, which likely significantly affected the result quality. The Tables C.1, C.2, C.3, C.4, C.5 and C.6 correspond to the examples shown in Figure C.5. In the tables we show the expectation values of several observables M, obtained from the output of the VQLS and we compare them to the exact ones.

M	$\langle M\rangle_{\text{exact}}$	$\langle M\rangle_{\text{exp}}$	$D(M)^2$
Z	0	0.04 ± 0.02	0.002 ± 0.002

TABLE C.1: Expectation value of an observable M computed with the exact solution, and with the output solution of VQLS. $D(M)$ measures the difference between these two results. The linear system considered is $A_{2\times 2} = H$, and $|b\rangle = X\,|0\rangle$.

M	$\langle M\rangle_{\text{exact}}$	$\langle M\rangle_{\text{exp}}$	$D(M)^2$
Z	-1	-0.819 ± 0.005	0.032 ± 0.002

TABLE C.2: Expectation value of an observable M computed with the exact solution, and with the output solution of VQLS. $D(M)$ measures the difference between these two results. The linear system considered is $A_{2\times 2} = \mathbb{1} + 0.25\, Z$, and $|b\rangle = X\,|0\rangle$.

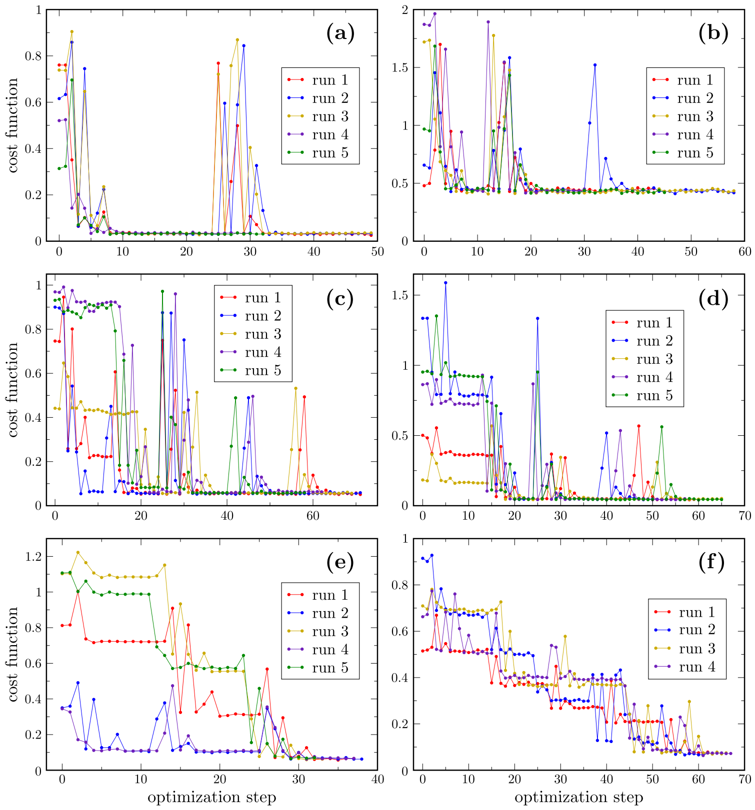

FIGURE C.5: Cost function versus number of optimization steps. The classical optimization algorithm employed is the Powell method which uses an unconstrained bi-directional search. Randomization in this algorithm occasionally leads to spikes in the cost function, visible in the plots, which quickly deteriorate as the optimizer reverts back towards better parameters. (a) $A_{2\times2} = H$, and $|b\rangle = X|0\rangle$. (b) $A_{2\times2} = \mathbb{1} + 0.25Z$, and $|b\rangle = X|0\rangle$. (c) $A_{4\times4} = X_1H_2$, and $|b\rangle = H_1H_2|\vec{0}\rangle$. (d) $A_{4\times4} = \mathbb{1} + 0.25Z_2$, and $|b\rangle = H_1|\vec{0}\rangle$. (e) $A_{8\times8} = \mathbb{1} + 0.25Z_3$, and $|b\rangle = H_1H_2|\vec{0}\rangle$. (f) $A_{32\times32} = \mathbb{1} + 0.25X_5$, and $|b\rangle = H^{\otimes5}|\vec{0}\rangle$.

M	$\langle M\rangle_{\text{exact}}$	$\langle M\rangle_{\text{exp}}$	$D(M)^2$
Z_2	1	0.943 ± 0.003	0.0032 ± 0.0003
Z_1	0	0.02 ± 0.04	0.000 ± 0.002
Z_1Z_2	0	0.02 ± 0.04	0.000 ± 0.002

TABLE C.3: Expectation value of observables M computed with the exact solution, and with the output solution of VQLS. $D(M)$ measures the difference between these two results. The linear system considered is $A_{4\times4} = X_1H_2$, and $|b\rangle = H_1H_2|\vec{0}\rangle$.

M	$\langle M\rangle_{\text{exact}}$	$\langle M\rangle_{\text{exp}}$	$D(M)^2$
Z_2	1	0.930 ± 0.004	0.0047 ± 0.0005
Z_1	0	0.00 ± 0.02	0.00000 ± 0.00006
Z_1Z_2	0	0.00 ± 0.02	0.00000 ± 0.00009

TABLE C.4: Expectation value of observables M computed with the exact solution, and with the output solution of VQLS. $D(M)$ measures the difference between these two results. The linear system considered is $A_{4\times4} = \mathbb{1} + 0.25Z_2$, and $|b\rangle = H_1|\vec{0}\rangle$.

M	$\langle M\rangle_{\text{exact}}$	$\langle M\rangle_{\text{exp}}$	$D(M)^2$
Z_3	1	0.88 ± 0.01	0.013 ± 0.002
Z_2	0	0.00 ± 0.04	0.0000 ± 0.0001
Z_2Z_3	0	0.00 ± 0.03	0.0000 ± 0.0002
Z_1	0	0.04 ± 0.05	0.002 ± 0.003
Z_1Z_3	0	0.04 ± 0.04	0.002 ± 0.004
Z_1Z_2	0	-0.002 ± 0.009	0.00000 ± 0.00004
$Z_1Z_2Z_3$	0	-0.004 ± 0.009	0.00002 ± 0.00009

TABLE C.5: Expectation value of observables M computed with the exact solution, and with the output solution of VQLS. $D(M)$ measures the difference between these two results. The linear system considered is $A_{8\times8} = \mathbb{1} + 0.25Z_3$, and $|b\rangle = H_1H_2\,|\vec{0}\rangle$.

M	$\langle M\rangle_{\text{exact}}$	$\langle M\rangle_{\text{exp}}$	$D(M)^2$
Z_5	0	0.180 ± 0.030	0.030000 ± 0.01000
Z_4	0	0.000 ± 0.100	0.000000 ± 0.00400
Z_4Z_5	0	0.000 ± 0.020	0.000000 ± 0.00040
Z_3	0	0.000 ± 0.100	0.000000 ± 0.00900
Z_3Z_5	0	0.000 ± 0.020	0.000000 ± 0.00040
Z_3Z_4	0	-0.006 ± 0.009	0.000000 ± 0.00010
$Z_3Z_4Z_5$	0	0.000 ± 0.001	0.000000 ± 0.000001
Z_2	0	0.100 ± 0.020	0.010000 ± 0.00500
Z_2Z_5	0	0.019 ± 0.009	0.000300 ± 0.00030
Z_2Z_4	0	0.000 ± 0.010	0.000000 ± 0.00004
$Z_2Z_4Z_5$	0	-0.001 ± 0.009	0.000000 ± 0.00003
Z_2Z_3	0	0.000 ± 0.010	0.000000 ± 0.00002
$Z_2Z_3Z_5$	0	0.005 ± 0.004	0.000020 ± 0.00004
$Z_2Z_3Z_4$	0	-0.007 ± 0.006	0.000050 ± 0.00009
$Z_2Z_3Z_4Z_5$	0	-0.002 ± 0.008	0.000000 ± 0.00004
Z_1	0	0.010 ± 0.020	0.000200 ± 0.00070
Z_1Z_5	0	0.005 ± 0.008	0.000030 ± 0.00009
Z_1Z_4	0	0.003 ± 0.005	0.000010 ± 0.00003
$Z_1Z_4Z_5$	0	-0.002 ± 0.007	0.000000 ± 0.00004
Z_1Z_3	0	0.000 ± 0.010	0.000000 ± 0.00001
$Z_1Z_3Z_5$	0	-0.002 ± 0.005	0.000000 ± 0.00002
$Z_1Z_3Z_4$	0	0.000 ± 0.008	0.000000 ± 0.00001
$Z_1Z_3Z_4Z_5$	0	-0.001 ± 0.004	0.000000 ± 0.00004
Z_1Z_2	0	0.006 ± 0.004	0.000040 ± 0.00005
$Z_1Z_2Z_5$	0	0.000 ± 0.010	0.000010 ± 0.00007
$Z_1Z_2Z_4$	0	-0.002 ± 0.001	0.000000 ± 0.00001
$Z_1Z_2Z_4Z_5$	0	0.002 ± 0.005	0.000000 ± 0.00002
$Z_1Z_2Z_3$	0	0.001 ± 0.007	0.000000 ± 0.00002
$Z_1Z_2Z_3Z_5$	0	0.000 ± 0.006	0.000000 ± 0.00001
$Z_1Z_2Z_3Z_4$	0	0.001 ± 0.002	0.000001 ± 0.000006
$Z_1Z_2Z_3Z_4Z_5$	0	0.002 ± 0.001	0.000004 ± 0.000007

TABLE C.6: Expectation value of observables M computed with the exact solution, and with the output solution of VQLS. $D(M)$ measures the difference between these two results. The linear system considered is $A_{32\times32} = \mathbb{1} + 0.25X_5$, and $|b\rangle = H^{\otimes5}\,|\vec{0}\rangle$.

M	$\langle M\rangle_{\text{exact}}$	$\langle M\rangle_{\text{exp}}$	$D(M)^2$
Z_5	0	0.1 ± 0.1	0.01 ± 0.04
Z_4	0	0.00 ± 0.04	0.0 ± 0.0005
Z_4Z_5	0	0.00 ± 0.01	0.0 ± 0.00007
Z_3	0	0.0 ± 0.1	0.0 ± 0.004
Z_3Z_5	0	0.01 ± 0.02	0.0002 ± 0.0009
Z_3Z_4	0	-0.002 ± 0.007	0.0 ± 0.00003
$Z_3Z_4Z_5$	0	0.000 ± 0.005	0.0 ± 0.000009
Z_2	1	0.971 ± 0.002	0.0008 ± 0.0001
Z_2Z_5	0	0.1 ± 0.1	0.01 ± 0.04
Z_2Z_4	0	0.00 ± 0.04	0.0 ± 0.0004
$Z_2Z_4Z_5$	0	0.00 ± 0.01	0.0 ± 0.00006
Z_2Z_3	0	0.0 ± 0.1	0.0 ± 0.003
$Z_2Z_3Z_5$	0	0.01 ± 0.02	0.0002 ± 0.0009
$Z_2Z_3Z_4$	0	-0.002 ± 0.006	0.0 ± 0.00003
$Z_2Z_3Z_4Z_5$	0	0.001 ± 0.005	0.0 ± 0.00001
Z_1	0	-0.02 ± 0.04	0.0 ± 0.001
Z_1Z_5	0	0.000 ± 0.007	0.0 ± 0.0006
Z_1Z_4	0	0.000 ± 0.004	0.0 ± 0.0000006
$Z_1Z_4Z_5$	0	0.000 ± 0.006	0.0 ± 0.00001
Z_1Z_3	0	-0.002 ± 0.006	0.0 ± 0.00003
$Z_1Z_3Z_5$	0	-0.001 ± 0.004	0.0 ± 0.00001
$Z_1Z_3Z_4$	0	0.004 ± 0.005	0.00001 ± 0.00004
$Z_1Z_3Z_4Z_5$	0	0.003 ± 0.004	0.00001 ± 0.00003
Z_1Z_2	0	-0.02 ± 0.03	0.0 ± 0.001
$Z_1Z_2Z_5$	0	0.000 ± 0.008	0.0 ± 0.0000004
$Z_1Z_2Z_4$	0	0.000 ± 0.005	0.0 ± 0.000004
$Z_1Z_2Z_4Z_5$	0	0.000 ± 0.006	0.0 ± 0.000006
$Z_1Z_2Z_3$	0	-0.001 ± 0.007	0.0 ± 0.00002
$Z_1Z_2Z_3Z_5$	0	0.000 ± 0.004	0.0 ± 0.000008
$Z_1Z_2Z_3Z_4$	0	0.004 ± 0.003	0.00001 ± 0.00003
$Z_1Z_2Z_3Z_4Z_5$	0	0.004 ± 0.003	0.00001 ± 0.00003

Table C.7: Expectation value of observables M computed with the exact solution, and with the output solution of VQLS. $D(M)$ measures the difference between these two results. The linear system considered is $A_{32\times 32} = \mathbb{1} + 0.2X_1Z_2 + 0.2X_1$, and $|b\rangle = H_1H_3H_4H_5\,|\vec{0}\rangle$.

D

Generative modeling with a noisy simulation

We have performed a noise simulation of the style-qGAN presented in Chapter 6 on an IBM Q device, taking as a device baseline the `ibmq_santiago` 5-qubit Falcon r4L quantum processor that we have used for our runs on real IBM Q hardware.

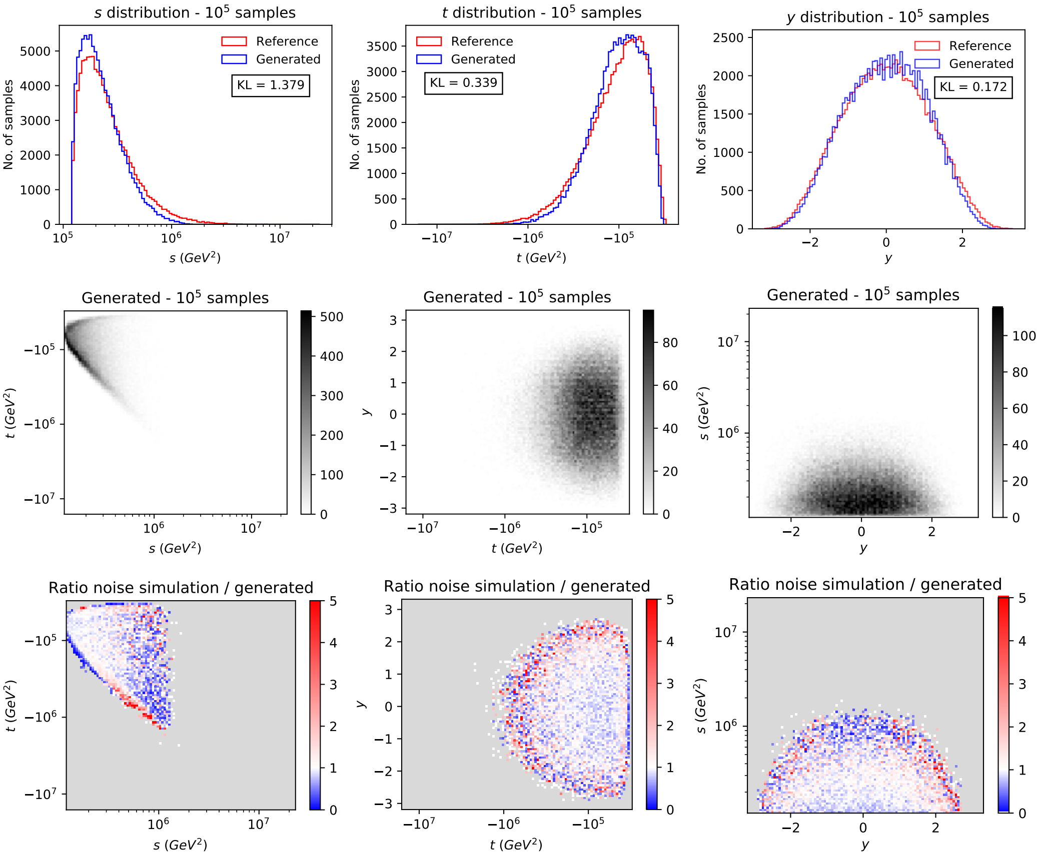

Figure D.1: Marginal samples distributions for the physical observables s, t, y in $pp \rightarrow t\bar{t}$ production at the LHC using the style-qGAN generator model in a noise simulation of `ibmq_santiago` device (top row), trained with 10^4 samples (top row), together with the corresponding two-dimensional sampling projections (middle row) and the ratio to the reference underlying prior MC distribution (bottom row). Note that we choose a grey background for the plots at the bottom row to highlight when the reference and generated samples are identical.

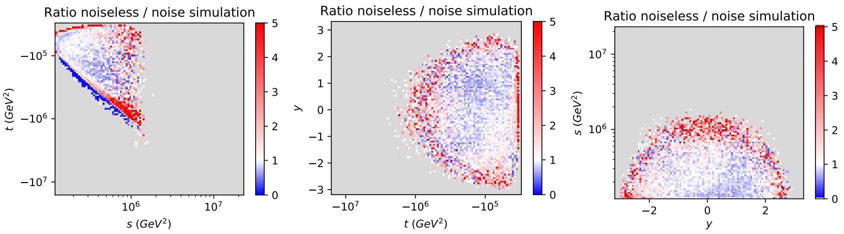

FIGURE D.2: Ratio of two-dimensional sampling projections using the noise simulation of `ibmq_santiago` device to the corresponding noiseless simulation.

The noise model takes into account the readout error probability of each qubit (mean value of the probability of reading $|1\rangle$ while being in the state $|0\rangle$, and the probability of reading $|0\rangle$ while being in the state $|1\rangle$), the relaxation time constants of each qubit (relaxation time and dephasing time), the gate error probability of each basis gate, and the gate length (timing of the gate) of each basis gate. The values are taken from the calibration information of the selected device for the noise simulation. Note that this calibration is performed at regular intervals. The generation of 10^5 samples on the actual machine took about one week, implying that the calibration parameters may have varied significantly during the full run.

We show in Figure D.1 the result of the noise simulation. The KL distances displayed in the top row are comparable to the KL distances reported in Figure 6.7. We also compare our noise simulation to the run on actual IBM Q hardware, the latter being reported in Section 6.6; this is shown in the bottom row. The plots show a significant amount of white points, signalling that the noise simulation seems to capture most of the errors induced by running on actual quantum hardware and that errors beyond the parameters reported in the previous paragraph are subdominant.

We also compare our noise simulation to the noiseless simulation. The results shown in Figure D.2 indicate that while the noise has an impact, as expected, there are still many points close to the ratio of one; therefore the style-qGAN still performs fairly well in a noisy environment. This led us to believe that running on actual quantum hardware will give reasonable results.