Investigation of quantum computers for quantum simulation and machine learning

Thesis by
Hirsh Kamakari

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2024
Defended March 13, 2024

Hirsh Kamakari
ORCID: 0000-0002-5377-9631

ACKNOWLEDGEMENTS

I am deeply thankful to my advisor, Prof. Austin J. Minnich, for his guidance and support throughout my Ph.D. His steadfast commitment to the development of his students has helped me grow both in terms of research and in the communication of my work. His dedication to creating a positive research environment both within the group and with collaborators has allowed me to freely explore my research interests.

I am thankful to Prof. Andrei Faraon, Prof. Xie Chen, and Prof. Manuel Endres for serving on my defense committee.

I am also grateful to Sonya, Kristen, Lynn, Jennifer, and Christy, our department, options, and group administrators for taking care of all essential non-research duties.

I am very thankful to my collaborators, Mario, Tanvi, Ryan, Abhinav, and Yaodong, for their mentorship and invaluable discussions and contributions to the projects we worked on together.

I am infinitely grateful for the friends I have made during my Ph.D. journey. They will remain unnamed just in case I forget anyone. Without their friendship I would not have discovered so many places in and around Pasadena, California, the US, and Mexico. They have all brought me lots of laughter and good times, and meeting all of them has been one of the main highlights of my five years at Caltech.

Finally, I would like to thank my family for supporting me from the moment I was born all the way to the writing of this sentence and beyond. This journey would not have been possible without them.

ABSTRACT

The use of quantum mechanical phenomena for information processing has the potential to solve computational problems which are believed to be intractable for classical computers. Inspired by this potential, the last several decades has seen rapid development in both the theory and practice of quantum information processing. In this thesis, we explore three applications of quantum computing for the physical and computational sciences.

The first potential application is for the simulation of open quantum systems. We introduce two algorithms for the simulation of open quantum systems governed by a Lindblad equation. Based on adaptations of the quantum imaginary time evolution algorithm, these methods transform non-unitary open system evolution into unitary evolution which can be implemented on contemporary quantum hardware. We demonstrate these algorithms on IBM's quantum hardware via the simulation of the spontaneous emission of a two-level system and the dissipative transverse field Ising model.

Next, we explore efficient methods to probe measurement induced phase transitions using superconducting circuits. These phase transitions occur in monitored quantum systems as the measurement rate of randomized single qubit measurements increases. We overcome two exponential bottlenecks which limited the system sizes of previous experiments on superconducting circuits by employing a cross-entropy benchmarking protocol and Clifford based circuit compression techniques. We observed measurement induced phase transitions on systems of up to 22 physical qubits.

Finally, we switch our attention to machine learning, where we prove rigorous quantum advantages for adversarially robust classification. By constructing a learning task based on widely accepted cryptographic assumptions, we show a necessary condition for the utility of quantum computers for robust classification. In particular, we show that for the learning task we construct, any efficient classical learner cannot robustly classify better than chance, whereas a quantum learner can efficiently and robustly classify data with high accuracy.

Through these studies, we show that quantum computers have potential application in the physical and information sciences in both the near and long term.

PUBLISHED CONTENT AND CONTRIBUTIONS

1. Hirsh Kamakari et al. Digital quantum simulation of open quantum systems using quantum imaginary–time evolution. In: *PRX Quantum* 3.1 (2022), p. 010320. https://doi.org/10.1103/PRXQuantum.3.010320.

   Contributions: H.K. conceptualized the project, designed and optimized the quantum circuits, carried out the simulation runs on the quantum computers, performed simulations and analyzed the experimental data, and wrote the manuscript.

2. Hirsh Kamakari et al. Experimental demonstration of scalable cross-entropy benchmarking to detect measurement-induced phase transitions on a superconducting quantum processor. In: *arXiv preprint arXiv:2403.00938* (2024). https://arxiv.org/abs/2403.00938.

   H.K. designed and optimized the quantum circuits, carried out the simulation runs on the quantum computers, performed simulations and analyzed the experimental data, and wrote the manuscript.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Number *Page*

LIST OF TABLES

Number *Page*

Chapter 1

INTRODUCTION

1.1 Early history of quantum computing

Quantum mechanics as a resource for information processing was first motivated in the early 1980's as a solution to the classically hard problem of simulating interacting quantum systems [1, 2, 3]. By 1985, quantum computing was formalized in terms of quantum Turing machines [4, 5] followed by the circuit model of quantum computation [6], which were subsequently proven to be equivalent to each other in 1993 [7]. The circuit model of quantum computation led to many rapid advances in the development of quantum algorithms which exhibit reduced time or space complexity over their classical counterparts [8, 9, 10, 11, 12]. In the late 1990s, the problem of simulating quantum systems with a quantum computer was further addressed and algorithms which allow for the efficient quantum computation of time dependent expectation values of various types of quantum mechanical systems were proposed [13, 14, 15, 16]. Inspired by the prospects of quantum computing for various applications, various proposals for qubits in different physical systems were introduced, including in nuclear magnetic resonance (NMR) systems [17, 18], trapped ions [19], quantum dots [20], and superconducting circuits [21]. Experimental realizations quickly followed, with the first quantum gates and algorithms implemented in trapped ion and NMR systems [22, 23]. In the following decades, exponential progress was made in key metrics for realizing qubits for quantum computation, such as coherence times, readout error rates, and gate errors [24, 25, 26]. The rapid and exciting progress of quantum computing over the last 40 years has spurred many investigations on the utility of quantum computing for applications ranging from the simulation of physical systems to cryptography to machine learning among many others.

1.2 Current state and prospects of quantum computing

In this section we briefly summarize the current state of quantum computing, including recent advancements, challenges, and its near-term prospects. We discuss the current state of experimental realizations of quantum computers and applications across various fields.

Various platforms for quantum computing have emerged over the past several decades, including superconducting qubits, trapped ions, solid state platforms, and neutral atoms. Coherence times exceeding one millisecond and single-qubit gate fidelities of 0.99991(1) have recently been obtained in superconducting qubits [27] and two-qubit gate fidelities exceeding 99.5% have also been demonstrated [28, 29]. Although long coherence times and high two-qubit gate fidelities are certainly achievable in superconducting qubits, it is important to note that these results were obtained in small systems only, and scaling up to larger systems may reduce these metrics. Higher two-qubit gate fidelities have been achieved in trapped ion systems, allowing for the demonstration of all the requirements of fault tolerant quantum computing [30].

Current research in applications of quantum computing includes cryptography, drug discovery, materials science, finance, and machine learning. The necessity to protect private information on the classical internet has spurred the development of quantum-safe encryption protocols. Any encryption scheme which can be broken by factoring or discrete logarithms, such as the RSA cryptosystem [31], can be broken by Shor's algorithm, although it would require billions of operations on thousands of logical qubits [32] and so is not under imminent threat. To prevent private information from being exposed in the long term, post-quantum cryptographic methods are being developed which would require super-polynomial time to break even with a quantum computer [32].

In the near term, applications of quantum computing to drug discovery and material science may provide the first benefits of quantum computing [33, 34, 35, 36]. Because these applications are fundamentally quantum mechanical in nature and additionally take place in noisy systems, noisy quantum computers may provide advantages of classical computers in computing fundamental properties of these systems as well as pathways for creating them. Quantum computers are expected to accelerate drug discovery by simulating molecular interactions, optimizing chemical reactions, and predicting drug properties with high accuracy.

Quantum applications for finance [37, 38, 39] and machine learning [40, 41, 42] are also active areas of research, although rigorous quantum advantages are still lacking.

In this thesis, we focus on several applications of quantum computing for quantum simulation, critical phenomena, and machine learning. To provide a foundation for the following chapters, in Section 1.3 we introduce the notion of circuit based quantum computing and several primitives such as quantum gates and measurements.

In Section 1.4 we introduce noise sources and the models used to describe them for the experimental demonstrations presented in this thesis. Finally, in Section 1.5 we outline the three main topics of this thesis, providing a motivation and the main achievements of each project. In the following chapters we delve into the technical details and results of each project. The final chapter presents a summary and outlook of the applications of quantum computing in the context of the results presented in this thesis.

1.3 Circuit based quantum computing

In this section, we present the fundamental building blocks of quantum computing. This includes the basic unit of quantum information, the qubit, as well as operations which can be applied to systems of qubits in order to carry out a quantum computation. Although there are several formalisms used to describe quantum computation, such as measurement based quantum computing [43], topological quantum computing [44], the quantum Turing machine [4, 5] and the circuit model [6], they are all equivalent in computational power: any model can efficiently simulate any other model. By far the most common and simplest formalism is the circuit model, which we describe in this section. All of the information in the section can be found in Reference [45].

Qubits and where they live

The fundamental unit of discrete quantum computation is the qubit. We define a qubit as an element of a two-dimensional complex Hilbert space $\mathcal{H}_2$. To make concrete calculations, we define an orthonormal basis for the Hilbert space $\{|0\rangle, |1\rangle\}$. In column vector notation, we define $|0\rangle = (1\ 0)^\top$ and $|1\rangle = (0\ 1)^\top$. Operations that we can apply to qubits consists of unitary operations, elements of the unitary group $U(\mathcal{H}_2)$, and projective measurements defined by two-dimensional projective operators P, satisfying $P^2 = P$. In general, the operations we can apply to qubits are completely positive trace preserving (CPTP) maps and positive operator-valued measures (POVMs), which include the aforementioned unitary evolutions and projective measurements; however, for the purpose of this thesis we do not need such general operations and for simplicity we omit the details of CPTP maps and POVMs.

The Hilbert space of n qubits is defined as the tensor product of the Hilbert space of each qubit, $\mathcal{H} = \mathcal{H}_2^1 \otimes \mathcal{H}_2^2 \cdots \otimes \mathcal{H}_2^n$, which has dimension 2^n. Operations we can apply to systems of qubits are the unitary operations and projective measurements on the total Hilbert space. n qubit observables are defined as 2^n-dimensional Hermitian

operators. A general n qubit quantum state $|\psi\rangle$ can be always be represented as

$$|\psi\rangle = \sum_{x=0}^{2^n-1} a_x |x\rangle, \tag{1.1}$$

where unit probability dictates that $\sum_x |a_x|^2 = 1$ and each x is written in its binary representation.

Quantum circuit diagrams

The previous subsection described the abstract mathematics of qubits and the operations which can be applied to them. In order to develop algorithms and applications for quantum computers, a more concrete description of the allowable operations is required. This is provided by the circuit model of quantum computation, which we describe here. Figure 1.1 shows an example of a circuit diagram containing multi-qubit unitaries, classical feed-forward, and projective measurements.

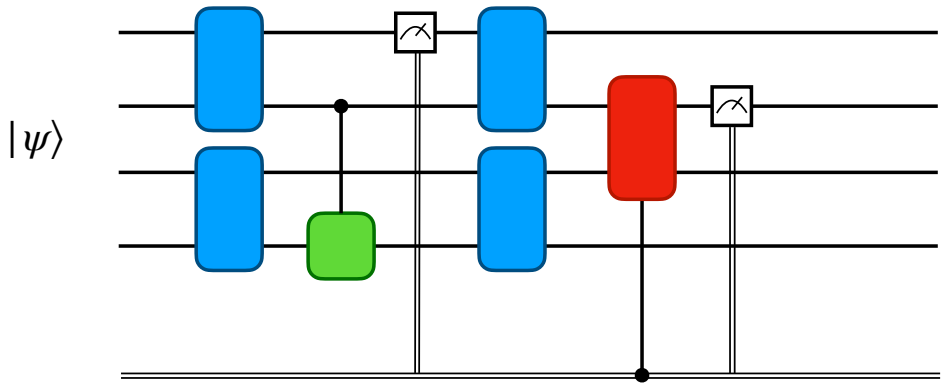

Figure 1.1: Quantum circuit diagram for a computation with 4 qubits. The initial state is denoted by $|\psi\rangle$. The blue rectangles represents 2 qubit unitaries, the metered box represents a measurement, and the red controlled rectangle represents a 2 qubit unitary conditioned on a measurement result. The green controlled square represents a controlled single qubit gate. The double line at the bottom of the circuit represents the classical bits used in the computation.

We can break up the circuit diagram into five main sections, which reads from left to right with time increasing to the right. On the left, we write the initial state which is used for the quantum computation. The bulk of the circuit consists of three sections, which in order of increasing time are: 1. state preparation, 2. the computation,

and 3. measurement preparation. The final section of the quantum circuit, shown on the right of Figure 1.1, consists of measurements in the computational basis. We typically assume that the input state is a fixed product state, typically (and arbitrarily) taken to be $|0\rangle^{\otimes n}$. The state preparation step then takes the initial state to the desired input state required for the computation, which is then followed by the operations required for the main part of the computation. The measurement preparation step allows us to measure in bases other than the computational basis. Finally, the computation concludes with measurement of the qubits, which yields a classical n-bit string. To obtain the expectation value of observables, we need to run the circuit multiple times, each time obtaining a different bitstring. We average the outputs of all the runs of the circuit in order to get the expectation value.

Unitary operations as quantum gates

For a general quantum computation, we would like to be able to apply an n-qubit unitary operation; however, due to physical constraints, contemporary quantum computers typically only allow for two-qubit unitaries, although multi-qubit unitaries are an active area of research [46, 47]. The set of physically implementable unitaries (for a given physical system) is known as a gate set, and elements of the gate set are the quantum gates we can apply to the system. In order to carry out a computation when physical restrictions only allow for two-qubit gates, we need to decompose n-qubit unitaries into a sequence of two-qubit unitaries. As shown in [45], any unitary operation can can be decomposed into a sequence of two-qubit unitaries. Additionally, almost any two-qubit gate is universal for quantum computation, provided that we can apply any single qubit gate [48]. A gate set which is universal for quantum computation is a gate set which allows for approximation of any multi-qubit unitary operation to arbitrary precision. Although the group $SU(\mathcal{H}_2)$ is a continuous group, any single qubit unitary can be approximated to arbitrary precision provided we have access to a single qubit gate set which generates a dense subgroup of $SU(\mathcal{H}_2)$ [49].

Some important single qubit gates which will be used throughout this thesis are the single qubit Pauli operators

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \text{and} \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{1.2}$$

The Pauli operators are sometimes also denoted by $\sigma_x = X$, $\sigma_y = Y$, and $\sigma_z = Z$. Single qubit rotations around the x, y, and z axis are generated by exponentiating

the respective Pauli operators:

$$R_x(\theta) = \exp(-i\theta\sigma_x/2), \quad R_y(\theta) = \exp(-i\theta\sigma_y/2) \text{ and } R_z(\theta) = \exp(-i\theta\sigma_z/2). \tag{1.3}$$

The three other single qubit gates which will also be used throughout this thesis are

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, \quad S = \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}, \quad \text{and} \quad T = \begin{pmatrix}1 & 0\\ 0 & \sqrt{i}\end{pmatrix}, \tag{1.4}$$

known as the Hadamard, Phase, and T gates, respectively.

Commonly used two-qubit entangling gates are the CNOT (or CX) gate, and the CZ gate, which have matrix representations

$$CX = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix} \quad \text{and} \quad CZ = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & -1\end{pmatrix}. \tag{1.5}$$

Projective measurements

The other primitive operation which is commonly used in quantum computing and in this thesis is the single qubit projective measurement. Due to physical constraints, many quantum computing systems are only able to measure single qubits in the computational basis. We represent these single qubit projective measurements by their projectors,

$$P_0 = \begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix} = \frac{1+Z}{2} \quad \text{and} \quad P_1 = \begin{pmatrix}0 & 0\\ 0 & 1\end{pmatrix} = \frac{1-Z}{2}. \tag{1.6}$$

For a single qubit state $|\psi\rangle = a|0\rangle + b|1\rangle$, measuring the qubit in the computational basis yields 0 or 1 probabilistically, with the probability of measuring either outcome equal to

$$\mathbb{P}[0] = \langle\psi|P_0|\psi\rangle = |a|^2 \quad \text{and} \quad \mathbb{P}[1] = \langle\psi|P_1|\psi\rangle = |b|^2. \tag{1.7}$$

Following the measurement, the state is projected into the subspace spanned by the projection operator corresponding to the measurement outcome:

$$|\psi\rangle \to \frac{P_x|\psi\rangle}{\langle\psi|P_x|\psi\rangle} \tag{1.8}$$

if the outcome of the measurement is $x \in \{0, 1\}$.

For measurements not in the computational basis, a unitary operation is first required to rotate the computational basis to the desired measurement basis. For example,

to measure in the x basis, we first apply a Hadamard gate to the qubit followed by a measurement in the computational basis.

Finally, we note that an n qubit Pauli measurement can always be broken down into a sequence of n CNOT gates, single qubit gates, and one single-qubit projective measurement in the computational basis.

1.4 Noise models for contemporary quantum computers

In the previous section we described the basic components of quantum circuits. In an ideal physical implementation, all the necessary operations can be carried out perfectly; however, in real-world physical systems, the execution of these operations inevitably introduces some degree of noise. In this section, we introduce the basic theory of noise processes in quantum computers and describe ways to carry out computations in the presence of noise.

Representing noise and errors with quantum channels

Although there are many microscopic mechanisms which induce noise in qubits [50, 51, 52], all noise processes can be described using quantum channels. A quantum channel is defined as a completely positive, trace preserving (CPTP) map from an initial state ρ to a final state $\Phi(\rho)$. We can always decompose a CPTP map Φ into a basis of operators in the form

$$\Phi(\rho) = \sum_i M_i \rho M_i^\dagger, \tag{1.9}$$

where the M_i's, known as Kraus operators, satisfy the relation

$$\sum_i M_i M_i^\dagger = I. \tag{1.10}$$

The Kraus operators of commonly used quantum channels often have a intuitive physical interpretation. We describe here some common single qubit quantum channels which are used to describe various physical noise processes.

The amplitude damping channel is often used to model the spontaneous emission of a two-level system from the excited state to the ground state. If the probability of decaying from the excited state to the ground state is p, then the Kraus operators describing this channel are given by

$$M_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{pmatrix}, \quad M_1 = \begin{pmatrix} 0 & \sqrt{p} \\ 0 & 0 \end{pmatrix}. \tag{1.11}$$

Since relative phases are important to maintain throughout a computation, a noise channel describing the loss of phase coherence is also commonly used, known as a dephasing channel. The dephasing channel can be described using three Kraus operators:

$$M_0 = \sqrt{1-p}I, \quad M_1 = \begin{pmatrix} \sqrt{p} & 0 \\ 0 & 0 \end{pmatrix}, \quad M_2 = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{p} \end{pmatrix}, \tag{1.12}$$

where p. An alternate representation of the dephasing channel can be shown to be desrcibed by the two Kraus operators

$$M_0 = \frac{\sqrt{p}}{2}(1+Z), \quad M_0 = \frac{\sqrt{p}}{2}(1-Z), \tag{1.13}$$

which yields the channel

$$\Phi(\rho) = \left(1 - \frac{p}{2}\right)\rho + \frac{p}{2}Z\rho Z. \tag{1.14}$$

This channel has the interpretation that with probability $1 - p/2$ we do nothing to the state and with probability $p/2$ we apply a Pauli Z to the state. To see the effect of the dephasing channel, we can apply it to a qubit in an arbitrary pure state $|\psi\rangle = a|0\rangle + b|1\rangle$. The channel maps the density operator as

$$\Phi(|\psi\rangle\langle\psi|) = \Phi\begin{pmatrix} |a|^2 & ab^* \\ a^*b & |b|^2 \end{pmatrix} = \begin{pmatrix} |a|^2 & (1-p)ab^* \\ (1-p)a^*b & |b|^2 \end{pmatrix}. \tag{1.15}$$

We see that as p increases from 0 to 1, the off-diagonal elements (the phase information) are lost.

The dephasing channel is a specific example of the more general Pauli channel. We can interpret a Pauli channel as probabilistically applying different Pauli operators to a qubit. We write the map as

$$\Phi(\rho) = (1 - p_x - p_y - p_z)\rho + p_x X\rho X + p_y Y\rho Y + p_z Z\rho Z. \tag{1.16}$$

We interpret this channel as applying X with probability p_x, applying Y with probability p_y, applying Z with probability p_z, and doing nothing with probability $1 - p_x - p_y - p_z$. If we set $p_x = p_y = 0$, we obtain a dephasing channel, and if we set $p_x = p_y = p_z$ we obtain what is known as a depolarizing channel.

Coherent errors

The error channels in the previous section all describe stochastic processes which affect the qubits in a physical quantum computer. In addition to random processes,

there are also systematic coherent errors which can arise due to miscalibrated gates [53]. For example, if we intend to apply the gate $\exp(i\theta Z)$ to a qubit, due to imperfect experimental control we may over rotate and instead apply the gate $\exp(i(\theta+\delta)Z)$. This leads to the coherent error $\exp(i\delta Z)$. Importantly, it is believed that coherent errors will not cause significantly more damage than Pauli errors for fault tolerant quantum computing [54].

Measurement errors and readout error mitigation

Another source of error during a computation occurs at when the qubits are measured, also known as qubit readout. When reading out a qubit, a readout error results in measuring a qubit as being in the $|0\rangle$ state when it is actually in the $|1\rangle$ state and vice versa.

Readout error mitigation (ROEM) is a collection of techniques used to help reduce the impact of readout errors [55, 56, 57]. The simplest form of ROEM models measurement errors classically by first constructs a transition matrix

$$M = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix}, \tag{1.17}$$

where p_{xy} is the probability of measuring the qubit to be in state $|x\rangle$ given that the qubit is in state $|y\rangle$. During a computation, we typically repeat the computation multiple times. If we repeat a single qubit experiment N times, then N_0 times we will readout 0 and N_1 times we will readout 1, with $N_0 + N_1 = N$. We can correct for readout errors by applying the inverse of M to our raw results:

$$\begin{pmatrix} \tilde{N}_0 \\ \tilde{N}_1 \end{pmatrix} = M^{-1} \begin{pmatrix} N_0 \\ N_1 \end{pmatrix}. \tag{1.18}$$

We use our corrected results to calculate expectation values. More advanced techniques can be used to improve the fidelity of calculations further [56, 55].

1.5 Thesis outline

Here we give a high level overview of the three main topics of this thesis.

Digital quantum simulation of open quantum systems using quantum imaginary time evolution

In Chapter 2, we present two algorithms for the dynamical simulation of open quantum systems on contemporary quantum hardware and demonstrate their use for small open quantum systems of up to five qubits [58]. While many studies focus

on computing ground state properties or simulating unitary dynamics of closed systems, open quantum systems are an interesting target of study owing to their ubiquity and rich physical behavior. However, their non-unitary dynamics are also not natural to simulate on digital quantum devices.

We consider the dynamics of spin systems governed by a local Lindblad equation. Using two adaptations of the quantum imaginary time evolution (QITE) algorithm [59], we develop two digital quantum algorithms to simulate the systems dynamics. We demonstrate the algorithms on IBM Quantum's hardware with simulations of the spontaneous emission of a two level system and the dissipative transverse field Ising model.

Experimental demonstration of measurement induced phase transitions using cross-entropy benchmarking

In Chapter 3, we consider a relative of traditional open quantum systems, so called monitored quantum systems. These are quantum systems which, while undergoing unitary evolution, are also measured by an outside observer. Such quantum systems are predicted to host novel non-equilibrium phases and quantum information phase transitions. Yet, experimental observations of these phenomena have faced both fundamental and technical challenges [60]. In Chapter 3, we study experimental realizations and efficient probes of such phase transitions in prototypical hybrid circuit models on IBM's superconducting processors containing up to 22 qubits within 8 qubit-hours. A combination of cross entropy benchmark [61] and circuit compression techniques [62] allows us to minimize the effect of uncontrolled device noise, and access system sizes of $L \leq 44$ without exponential overheads associated with post-selection of quantum trajectories, error mitigation, or quantum state tomography. From experimental data we extract critical exponents comparable to theoretical predictions in systems with one-dimensional and all-to-all connectivities.

Quantum advantage in adversarially robust machine learning

In Chapter 4 we shift our focus to applications of quantum computing to problems in machine learning. With the widespread deployment of modern machine learning, robustness to adversarial attacks, attacks in which a malicious adversary can perturb the data, are critical. In response, significant effort has been invested into both developing methods for adversarial attacks as well as mechanisms to defend against such attacks. Although quantum computing has emerged as a potential resource for machine learning, much less attention has been paid to the question of

adversarial robustness than in the classical case. This raises the natural question of whether quantum learning algorithms offer any advantages in robustness over classical learning algorithms. In Chapter 4, we answer the question in the affirmative by constructing a learning task which is 1. easy to learn non-robustly, 2. hard to learn robustly for any classical learner, and 3. easy to learn robustly for a quantum learner. While this task is not practically relevant, it provides a proof of principle that quantum computing offers advantages in robust machine learning tasks and we hope it stimulates further work on adversarial robustness of quantum machine learning algorithms.

Finally, in Chapter 5, we summarize the results of this thesis and discuss further directions of study related to our findings.

Chapter 2

DIGITAL QUANTUM SIMULATION OF OPEN QUANTUM SYSTEMS USING QUANTUM IMAGINARY TIME EVOLUTION

[1] Hirsh Kamakari et al. Digital quantum simulation of open quantum systems using quantum imaginary–time evolution. In: *PRX Quantum* 3.1 (2022), p. 010320. URL: https://doi.org/10.1103/PRXQuantum.3.010320.

One of the first proposed applications for quantum computing was for the simulation of quantum many-body systems [1, 2, 3]. By the late 90's quantum simulation algorithms for various types of systems had been proposed [13, 14, 15, 16]. The first quantum simulation algorithms focused either on simulating the dynamics of closed quantum systems or on finding ground and low lying excited states. Open quantum systems, quantum systems which interact with an environment, can be relatively challenging to simulate since the effect of the environment also needs to be incorporated into the simulation. In particular, environmental influence leads to non-unitary evolution which is not natural to implement on quantum computers. In this chapter, we present two algorithms for the dynamical simulation of open quantum systems governed by a Lindblad equation [63]. Both algorithms use an adaptation of the quantum imaginary time evolution algorithm [59]. The first algorithm we present uses a vectorized time evolution to implement the dynamics of the density operator on a quantum computer. The vectorization process leads to an ancilla overhead equal to the number of qubits of the corresponding closed quantum system to be simulated. The second algorithm does not require any ancilla qubits, but instead requires sampling over multiple initial states. We present a comparison of the resource requirements of both algorithms as well as experimental demonstrations of both algorithms for quantum systems of up to five qubits.

2.1 Introduction

The development of quantum algorithms to simulate the dynamics of quantum many-body systems is now a topic of interest owing to advances in quantum hardware [64, 65, 66]. While the real-time evolution of closed quantum systems on digital quantum computers has been extensively studied in the context of spin models [67, 68, 69, 70, 71, 72, 73], fermionic systems [74, 75], electron-phonon interactions [76], and

quantum field theories [77, 78, 79], fewer studies have considered the time evolution of open quantum systems, which exhibit rich dynamical behavior due to coupling of the system to its environment [80, 63]. However, this coupling leads to non-unitary evolution which is not naturally simulable on quantum hardware.

Early approaches to overcome this challenge included use of the quantum simulators' intrinsic decoherence [81] and direct simulation of the environment [82, 83, 84]. Theoretical works examined the resources required for efficient quantum simulation of Markovian dynamics [85, 86, 87], concluding that arbitrary quantum channels can be efficiently simulated by combining elementary quantum channels. Recently, several algorithms have been proposed for the digital quantum simulation of open quantum systems on the basis of the Kraus decomposition of quantum channels [88, 89, 90, 91, 92, 93] as well as variational descriptions of general processes to simulate the stochastic Schrödinger equation [71, 64] and the Lindblad equation [94]. Recently, explicit Trotterization of the Lindblad equation was used to simulate damping and dephasing of a single qubit using an additional ancilla qubit [95].

Simulation via Kraus decomposition is convenient when the Kraus operators corresponding to the time evolution of the system are known, such as modelling decoherence with amplitude damping or depolarizing channels. However, determining the Kraus operators of a general system requires either computing the full unitary evolution of both the system and environment or casting a master equation into an operator sum representation for the density operator. The latter procedure can be approximated analogously to Trotterization [93, 91] but requires either reset of ancillae qubits or a qubit overhead which scales linearly with the number of time steps in the simulation. Exactly determining the Kraus operators from the Lindblad equation is a classically hard task which is equivalent to solving the master equation [96] and so can only be applied to small systems. Explicit Trotterization circumvents the need to determine the Kraus operators representing the time evolution but has the same ancilla qubit overhead as in as the Kraus decomposition methods. Variational approaches [71, 64, 97] offer an alternative for simulating open system dynamics, but as in the case of closed systems require an ansatz and a potentially high dimensional classical optimization which is an NP-hard problem [98]. A quantum simulation of the stochastic Schrödinger equation was emulated in Ref. [71]. In this case, the quantum jumps, or discontinuous changes in the quantum state, was implemented via variational matrix-vector multiplication, thus incurring the disadvantages previously mentioned for variational approaches.

The common feature of the above algorithms is that they reformulate non-unitary open system dynamics into unitary dynamics which can be simulated on a quantum computer. A similar approach is used in variational approaches to imaginary time evolution [99] and the quantum imaginary time evolution (QITE) algorithm, which has recently been introduced as a way to prepare ground states and compute thermal averages on near-term devices [59]. QITE has since been used to compute finite-temperature correlation functions of many-body systems [100], scattering in the Ising model [101], and binding energies in quantum chemistry [102, 103] and nuclear physics [103]. It is therefore natural to consider how QITE might be adapted for open quantum system evolution.

In this chapter, we introduce two quantum algorithms to simulate open quantum dynamics using adaptations of the QITE algorithm and demonstrate them on IBM Quantum hardware. The first algorithm casts the Lindblad equation for the density operator into a Schrödinger-type equation with a non-Hermitian Hamiltonian. Time evolution is then achieved by simulating the unitary evolution via Trotterization, corresponding to the Hermitian component of the Hamiltonian and using QITE to simulate the anti-Hermitian component of the Hamiltonian. The second algorithm expresses the density operator in terms of an ansatz which is preserved during both real and imaginary time evolution. We demonstrate these algorithms on IBM Quantum hardware for two cases: the spontaneous emission of a two level system (TLS) in a heat bath at zero temperature, and the dissipative transverse field Ising model (TFIM) on two sites. We observe good agreement between the exact and hardware results, showing that the dynamics of open quantum systems are accessible on near-term quantum hardware.

The dynamics of a Markovian open quantum system can be described by the Lindblad equation

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = -i[H,\rho] + \sum_k \left(L_k \rho L_k^\dagger - \frac{1}{2}\{L_k^\dagger L_k, \rho\} \right) \tag{2.1}$$

where ρ is the density operator of the system, H is the system's Hamiltonian, and L_k are operators describing the coupling to the environment. The master equation in Lindblad form is often derived assuming weak coupling between system and environment and absence of memory effects (Born-Markov approximation) [104, 63].

We present two algorithms to simulate the master equation in Lindblad form on a digital quantum computer. The first quantum algorithm, based on a vectorization of the density operator, is described in Sec. 2.2; the second algorithm, which combines a QITE adaptation with an ansatz for the time-dependent density operator, is presented in Sec. 2.3.

2.2 Algorithm I

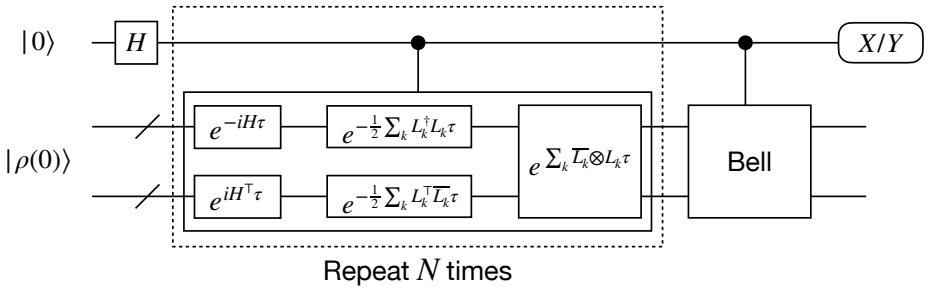

Figure 2.1: Time evolution for the vectorized density operator $|\rho\rangle$ (Algorithm I). $e^{-iH_1\tau}$ is a unitary operator and can be directly implemented on the quantum simulator. The non-unitary terms $e^{-\frac{1}{2}\sum_k L_k^\dagger L_k \tau}$ and $e^{\sum_k \overline{L_k}\otimes L_k \tau}$ are implemented via QITE. The unitary labelled "Bell" represents a unitary preparing the generalized $2n$-qubit Bell state. The / denotes a bundle of n qubits.

The Lindblad equation can be rewritten as a Schrödinger-type equation with a non-Hermitian Hamiltonian by transforming the $2^n \times 2^n$ density operator ρ into an 4^n component vector $|\rho\rangle$ by column stacking the density operator [105]. The resulting transformation of the Lindblad equation is

$$\frac{\mathrm{d}|\rho\rangle}{\mathrm{d}t} = \left[-i\mathbb{I}\otimes H + iH^\top \otimes \mathbb{I} + \sum_k (\overline{L_k}\otimes L_k - \frac{1}{2}\mathbb{I}\otimes(L_k^\dagger L_k) - \frac{1}{2}(L_k^\top \overline{L_k})\otimes \mathbb{I})\right]|\rho\rangle \tag{2.2}$$

where the bar indicates entry-wise complex conjugation and $|\rho\rangle = |\rho(t)\rangle$ is the vectorized density operator [105]. Separating Eq. (2.2) into Hermitian and anti-Hermitian parts, the time evolution of the initial state can be written as:

$$|\rho(t)\rangle = \exp\left(-i(H_1 - iH_2)t\right)|\rho(0)\rangle = \\ \left[\exp\left(-iH_1\tau\right)\exp\left(-H_2\tau\right)\right]^N|\rho(0)\rangle + O\left(\tau^2 N\right) \tag{2.3}$$

where in the last equality we have Trotterized to first order with time step $\tau = t/N$, and H_1 and iH_2 are the Hermitian and anit-Hermitian components of the vectorized Hamiltonian, respectively. The first term $\exp\left(-iH_1\tau\right)$ is unitary and can implemented on a quantum simulator via Trotterization and standard quantum simulation techniques [45, 65, 13]. The term $\exp\left(-H_2\tau\right)$ is non-unitary and so cannot be directly applied to the quantum register. Instead, we implement it on a digital quantum simulator via analogy to quantum algorithms for imaginary time evolution [59].

Imaginary time evolution of the Schrödinger equation with Hamiltonian H is carried out formally by substituting $\beta = it$ into the real time propagator $\exp\left(-itH\right)$. This technique is typically used to find ground states $|\psi\rangle = \lim_{\beta\to\infty}|\phi(\beta)\rangle/|||\phi(\beta)\rangle||$, where $|\phi(\beta)\rangle = \exp\left(-\beta H\right)|\phi(0)\rangle$ and $|\phi(0)\rangle$ has non-zero overlap with a ground state. If we interpret H_2 as the Hamiltonian of a system in the extended Hilbert space, $\exp\left(-H_2\tau\right)$ is an imaginary time evolution operator generated by H_2. The full time evolution is then applied as sequence of real and imaginary time evolutions, as shown in Fig. 2.1.

We present a brief review of the QITE algorithm reported in Ref. [59] which is used as a subroutine in this work. The QITE algorithm represents normalized imaginary time evolution in terms of unitary evolution as:

$$\frac{e^{-\beta H}|\psi\rangle}{||e^{-\beta H}|\psi\rangle||} = e^{-iA}|\psi\rangle, \tag{2.4}$$

where H is the system Hamiltonian, β is the imaginary time, and A is a Hermitian operator. The operator A can be represented with real coefficients in a complete basis of Hermitian operators, typically chosen to be the Pauli strings σ_i over the qubits of the system:

$$A = \sum_i a_i\sigma_i. \tag{2.5}$$

For an imaginary time step β, the coefficients a_i are determined (up to order β^2) by the linear system $Sa = b$, with

$$\begin{cases} S_{ij} = \langle\psi|\sigma_i^\dagger\sigma_j|\psi\rangle, \\ b_i = \frac{-i}{\sqrt{c}}\langle\psi|\sigma_i^\dagger H|\psi\rangle \end{cases} \tag{2.6}$$

where $c = \langle\exp(-2\beta H)\rangle$ is the norm squared of the un-normalized imaginary time evolved state.

Once the desired time and state $|\rho(t)\rangle$ are reached, measurements of an observable O are obtained by evaluating the expectation value $\langle O\rangle(t) = \mathrm{Tr}(O\rho(t))$ as $\langle O^\dagger|\rho\rangle$. $|O\rangle$ is the vector obtained from column stacking the matrix representation of O and so only the matrix representation of O in the computational basis is needed for this step. Lindbladian evolution preserves $\mathrm{Tr}(\rho)$ whereas the algorithm preserves $\mathrm{Tr}(\rho^2) = \langle\rho|\rho\rangle$, meaning that the operator ρ obtained from matricizing $|\rho(t)\rangle$ is not strictly a density matrix. However, the final state can be renormalized to have unit trace as $\rho'(t) = \rho(t)/\mathrm{Tr}(\rho(t))$. In practice, we normalize the final expectation value of a given observable instead. The final physical observables are thus given by $\langle O\rangle/\mathrm{Tr}(\rho)$. Therefore, obtaining measurements of observables on the state requires evaluating both $\langle O\rangle$ and $\mathrm{Tr}(\rho)$ at each time step.

Both quantities $\langle O\rangle(t)$ and $\mathrm{Tr}(\rho(t))$ can be obtained using a Hadamard test circuit [106]. In particular, $\mathrm{Tr}(\rho)$ can be evaluated up to a prefactor of $2^{-n/2}P^{-1/2}$ as $\langle 0|V^\dagger U|0\rangle$, where U is the circuit that prepares $|\rho\rangle$, V prepares the generalized Bell state $|\beta\rangle = 2^{-n/2}\sum_x |x\rangle\otimes|x\rangle$, $|x\rangle$ are the computational basis states on n qubits, and P is the purity of the initial state. Preparing the $2n$ qubit Bell state requires n Hadamard and n CNOT gates. Assuming $|\rho\rangle = U|0\rangle$ for a unitary U with gate decomposition requiring u_1 and u_2 single-qubit and CNOT gates, respectively, the measurement of $\mathrm{Tr}(\rho)$ requires a circuit with $O(n + u_1)$ single-qubit gates, $O(n + u_1)$ CNOT gates, and $O(n + u_2)$ CCNOT gates.

Measurement of $k-$local observables can be carried out similarly. We assume here without loss of generality that the $k-$local observable O has support on the first k qubits. The vectorized state can then be written as

$$|\rho\rangle = P^{-1/2}\sum_{x_1,x_2,y_1,y_2}\rho_{x_1x_2y_1y_2}|x_1x_2y_1y_2\rangle \tag{2.7}$$

where x_1, y_1 and x_2, y_2 are length k and $(n-k)$ bit strings, respectively, and P is the purity of the initial state. Defining the state

$$|O^\dagger\rangle = \sum_{x_1 y_1 z} \frac{\overline{O_{x_1 y_1}}}{\sqrt{2^{n-k}\mathrm{Tr}\left(O^\dagger O\right)}} |x_1 z y_1 z\rangle, \tag{2.8}$$

where the over-bar indicates complex conjugation, the expectation value of O can be evaluated (up to a pre-factor) as

$$\langle O^\dagger|\rho\rangle = \sum_{x_1 y_1 z} \frac{O_{x_1 y_1}}{\sqrt{2^{n-k}}} \frac{\rho_{x_1 z y_1 z}}{\sqrt{P}} = \frac{\mathrm{Tr}\left(O\rho\right)}{\sqrt{2^{n-k}\mathrm{Tr}\left(O^\dagger O\right) P}}. \tag{2.9}$$

The state $|O^\dagger\rangle$ can be prepared as

$$U_{O^\dagger} V_{n-k} |0_k, 0_{n-k}, 0_k, 0_{n-k}\rangle = U_{O^\dagger} \left[\frac{1}{\sqrt{2^{n-k}}} \sum_z |0_k, z, 0_k, z\rangle \right] \tag{2.10}$$

where V_{n-k} prepares the $n-k$ generalized Bell state and $U_{O^\dagger}$ prepares the $2k$ qubit state

$$|O^\dagger\rangle = \sum_{x_1 y_1} \frac{\overline{O_{x_1 y_1}}}{\sqrt{\mathrm{Tr}\left(O^\dagger O\right)}} |x_1 y_1\rangle. \tag{2.11}$$

We then measure the un-normalized expectation value of O using the Hadamard test. Since the purity is conserved by the algorithm, all observables can be renormalized after the measurement. Assuming a decomposition of U into u_1 and u_2 single-qubit and CNOT gates, respectively, and V into v_1 and v_2 single-qubit and CNOT gates, the total overhead for measurement of observables (including the trace evaluation) is $O\left(n + u_1 + v_1\right)$ single-qubit gates, $O\left(n + u_1 + v_1\right)$ CNOT gates, and $O\left(n + u_2 + v_2\right)$ CCNOT gates.

2.3 Algorithm II

Algorithm I allows for efficient simulation of the full density operator for many physical systems characterized by local interactions; however, it requires a doubling of the number of qubits and an overhead of an ancilla and controlled operations for evaluating observables. In particular, the circuit required for measurements is too deep for near-term hardware. We therefore introduce a second algorithm based on the variational ansatz used to obtain the non-equilibrium steady states of

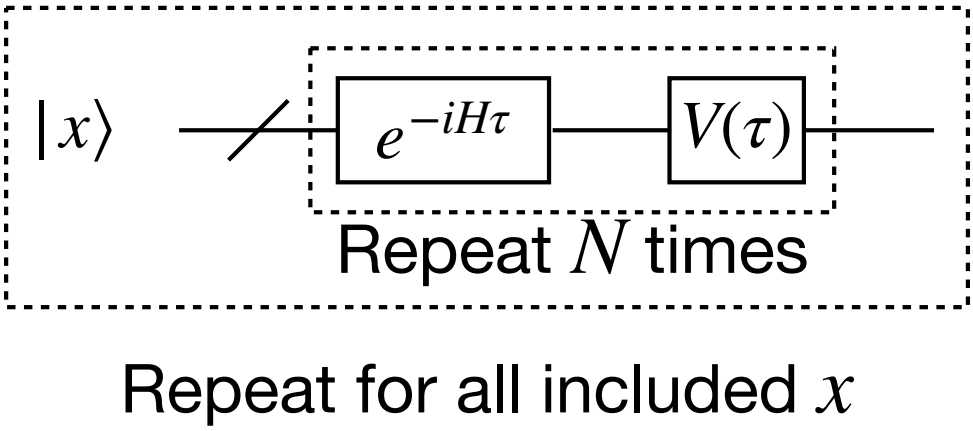

Figure 2.2: Time evolution for the purification-based algorithm (Algorithm II). x is a bit-string included in the index set I. $V(\tau)$ represents the non-unitary terms which need to be applied to the system for a time step τ. The / denotes a bundle of n qubits.

Markovian systems [97, 107] that overcomes these limitations. The isomorphism maps a density operator as

$$\rho = \sum_{x \in I} p_x U|x\rangle\langle x|U^\dagger \rightarrow |\rho\rangle = \sum_{x \in I} p_x U|x\rangle \otimes \overline{U}|x\rangle \tag{2.12}$$

where the $|x\rangle$'s label the n-qubit computational basis states and I is a subset of all 2^n possible bit-strings. In the rest of the paper the index set I is implied. We note that although we are using an ansatz for this algorithm, any density operator can be represented in this form provided the index set I is large enough. However, it should be noted that assuming polynomial resources to store the bit-string weights implies that the present algorithm employs a sparse approximation to represent the density matrix. The Lindblad master equation is mapped identically to the vectorization mapping, resulting in Eq. (2.2). In order to implement the resulting non-unitary time evolution on a quantum computer, we need to rewrite the time evolution operator in terms of products of unitary evolutions and imaginary time evolutions. The unitary evolution can be implemented directly on the quantum computer, while the imaginary time evolutions are implemented via a QITE adaptation, described in the following.

Derivation of the QITE linear system for complex time evolution

Here we derive the QITE linear systems which need to be solved to obtain the unitary time evolution of the density operator ansatz under vectorized Lindblad evolution. Consider $|\rho\rangle = \sum_x p_x T|x\rangle \otimes \overline{T}|x\rangle$, with T unitary. The complex time propagator is the same as in the vectorization method,

$$X(t) = \exp\left(\left[-i\mathbb{I}\otimes H + iH^\top \otimes \mathbb{I} + \sum_k (\overline{L_k}\otimes L_k - \frac{1}{2}\mathbb{I}\otimes(L_k^\dagger L_k) - \frac{1}{2}(L_k^\top \overline{L_k})\otimes\mathbb{I})\right] t\right). \tag{2.13}$$

Trotterizing results in

$$X(t) = \left\{\left[\exp\left(iH^\top\tau\right)\otimes\exp\left(-iH\tau\right)\right]\prod_k \exp\left(-\frac{L_k^\top\overline{L_k}\tau}{2}\right)^{\otimes 2}\exp\left(\overline{L_k}\otimes L_k\tau\right)\right\}^n + O\left(\tau^2 n\right) \tag{2.14}$$

with $\tau = t/n$. Using the identity $\overline{\exp(-iA)} = \exp(iA^\top)$ for A Hermitian and $\overline{\exp(B)} = \exp(\overline{B})$ for arbitrary B, the propagator can be rewritten as

$$X(t) = \left\{[U\otimes\overline{U}]\prod_k [V_k\otimes\overline{V_k}]W_k\right\}^n + O\left(\tau^2 n\right) \tag{2.15}$$

with $U := \exp(iH^\top\tau)$, $V_k := \exp(-L_k^\top\overline{L_k}\tau/2)$, and $W_k := \exp(\overline{L_k}\otimes L_k\tau)$. It is immediate that evolution with $U\otimes\overline{U}$ preserves the ansatz, as $(U\otimes\overline{U})\sum_x p_x T|x\rangle\otimes\overline{T}|x\rangle = \sum_x p_x(UT)|x\rangle\otimes(\overline{UT})|x\rangle$. The term $V\otimes\overline{V}$ also preserves the ansatz, but is an imaginary time evolution with Hamiltonian $L_k^\top\overline{L_k}/2$, and so requires a modified QITE algorithm, described below, for implementing $\exp(-L_k^\top\overline{L_k}\tau/2)T|x\rangle$. Due to the non-unitarity of V_k, we expect that in addition to a unitary evolution of the state, the weights p_x will also evolve in time. The final term, $W_k = \exp(\overline{L_k}\otimes L_k\tau)$, does not preserve the ansatz, and we use a modified version of QITE, described below, to effectively apply W_k while preserving the form of ansatz.

Implementing $V_k\otimes\overline{V_k}$ via a QITE adaptation

Under real time evolution by the non-unitary operator $V_k\otimes\overline{V_k}$, the evolution of $|\rho\rangle$ can be expressed as

$$V_k\otimes\overline{V_k}\sum_x p_x T|x\rangle\otimes\overline{T}|x\rangle = \sum_x (p_x+q_x)\exp(iA)T|x\rangle\otimes\exp(-i\overline{A})\overline{T}|x\rangle + O(\tau^2), \tag{2.16}$$

with $q_x \in \mathbb{R}$ and A a Hermitian operator with $\|A\|_2 = O(\tau)$. Defining $B := (1/2)L_k^\top \overline{L_k}$, we then have, to first order in τ,

$$\exp(-\tau B) \otimes \exp(-\tau \overline{B}) \sum_x p_x T|x\rangle \otimes \overline{T}|x\rangle = \sum_x (p_x + q_x) \exp(iA) T|x\rangle \otimes \exp(-i\overline{A})\overline{T}|x\rangle. \quad (2.17)$$

Expanding both sides to first order in τ and discarding higher order terms results in

$$-\tau \sum_x p_x (BT|x\rangle \otimes \overline{T}|x\rangle + T|x\rangle \otimes \overline{BT}|x\rangle) = \sum_x q_x T|x\rangle \otimes \overline{T}|x\rangle + i \sum_x p_x (AT|x\rangle \otimes \overline{T}|x\rangle - T|x\rangle \otimes \overline{AT}|x\rangle). \quad (2.18)$$

Taking the inner product of Eq. (2.18)) with $\langle y|T^\dagger \otimes \langle y|T^\top$ results in

$$\begin{aligned} -\tau \sum_x p_x (\langle y|T^\dagger BT|x\rangle \langle y|T^\top \overline{T}|x\rangle + \langle y|T^\dagger T|x\rangle \langle y|T^\top \overline{BT}|x\rangle) = \\ \sum_x q_x \langle y|T^\dagger T|x\rangle \langle y|T^\top \overline{T}|x\rangle + i \sum_x p_x (\langle y|T^\dagger AT|x\rangle \langle y|T^\top \overline{T}|x\rangle \\ - \langle y|T^\dagger T|x\rangle \langle y|T^\top \overline{AT}|x\rangle). \end{aligned} \quad (2.19)$$

Using the identities $\langle y|T^\dagger T|x\rangle = \langle y|T^\top \overline{T}|x\rangle = \delta_{xy}$ for T unitary results in

$$-\tau p_y (\langle y|T^\dagger BT|y\rangle + \langle y|T^\top \overline{BT}|y\rangle) = q_y + i p_y (\langle y|T^\dagger AT|y\rangle - \langle y|T^\top \overline{AT}|y\rangle). \quad (2.20)$$

Because A is Hermitian, we additionally have $\langle y|T^\dagger AT|y\rangle = \langle y|T^\top \overline{AT}|y\rangle$ so that the last two terms on the right-hand side cancel, resulting in

$$q_y = -2\tau p_y \mathrm{Re}\left[\langle y|T^\dagger BT|y\rangle\right]. \quad (2.21)$$

With the q_x's determined, we can now determine the operator A. Rearranging Eq. (2.18), we first isolate the terms containing A:

$$i \sum_x p_x (AT|x\rangle \otimes \overline{T}|x\rangle - T|x\rangle \otimes \overline{AU}|x\rangle) = -\tau \sum_x p_x (BT|x\rangle \otimes \overline{T}|x\rangle + T|x\rangle \otimes \overline{BT}|x\rangle) - \sum_x q_x T|x\rangle \otimes \overline{T}|x\rangle. \quad (2.22)$$

We define the right hand side as

$$|\Phi\rangle = -\tau \sum_x p_x (BT|x\rangle \otimes \overline{T}|x\rangle + T|x\rangle \otimes \overline{BT}|x\rangle) - \sum_x q_x T|x\rangle \otimes \overline{T}|x\rangle. \quad (2.23)$$

We then decompose A into a sum over Pauli strings with domain size D, $A = \sum_j a_j \sigma_j$, where the σ_j are Pauli strings acting on at most D qubits, $a_j \in \mathbb{R}$ and $a_j = O(\tau)$ for all j. Substituting into the left hand side of Eq. (2.22) yields

$$i \sum_{x,j} p_x a_j (\sigma_j T|x\rangle \otimes \overline{T}|x\rangle - T|x\rangle \otimes \overline{\sigma_j}\overline{T}|x\rangle) = \sum_j a_j |v_j\rangle = |\Phi\rangle, \tag{2.24}$$

where we have defined the vectors $|v_j\rangle := \sum_x p_x (\sigma_j T|x\rangle \otimes T^*|x\rangle - T|x\rangle \otimes \sigma_j^* T^*|x\rangle)$. Denoting by f the function

$$f(a) = \left\| |\Phi\rangle - i \sum_j a_j |v_j\rangle \right\|^2 = \\ \langle\Phi|\Phi\rangle + i \sum_j (a_j^* \langle v_j|\Phi\rangle - a_j \langle\Phi|v_j\rangle) + \sum_{jk} a_j^* a_k \langle v_j|v_k\rangle), \tag{2.25}$$

the optimal coefficients a_j are determined by minimizing f. This results in the set of equations

$$0 = \frac{\partial f}{\partial a_k} = -\mathrm{Im}\left[\langle v_k|\Phi\rangle\right] + \sum_j a_j \mathrm{Re}\left[\langle v_k|v_j\rangle\right]. \tag{2.26}$$

Defining the matrix $S_{jk} := \mathrm{Re}\left[\langle v_j|v_k\rangle\right]$ and the vector $b_j := \mathrm{Im}\left[\langle v_j|\Phi\rangle\right]$, the optimal coefficients a are the solution to the linear system $Sa = b$.

Using the definitions of $|v_j\rangle$ and $|\Phi\rangle$, we calculate calculate the matrix elements of S as

$$S_{jk} = \mathrm{Re}\left[\langle v_j|v_k\rangle\right] = \\ \sum_x p_x^2 \mathrm{Re}\left[\langle x|T^\dagger(\sigma_j\sigma_k + \sigma_k\sigma_j)T|x\rangle\right] - 2\sum_{xy} p_x p_y \mathrm{Re}\left[\langle x|T^\dagger\sigma_j T|y\rangle\langle x|T^\dagger\sigma_k T|y\rangle\right], \tag{2.27}$$

and the elements of b as

$$b_j = -2\tau \left(\sum_x p_x^2 \mathrm{Im}\left[\langle x|T^\dagger\sigma_j B T|x\rangle\right] + \sum_{xy} p_x p_y \mathrm{Im}\left[\langle x|T^\dagger\sigma_j T|y\rangle\langle y|T^\dagger B^\dagger T|x\rangle\right] \right). \tag{2.28}$$

Implementing W_k via a QITE adaptation

The real time evolution corresponding to $W_k = \exp(\tau \overline{L_k} \otimes L_k)$ can be determined completely analogously to that of $V_k \otimes \overline{V_k}$ above. The resulting equations are

$$\begin{cases} q_y = \tau \sum_x p_x \left|\langle y|T^\dagger L_k T|x\rangle\right|^2 \\ S_{jk} = \mathrm{Re}\left[\langle v_j|v_k\rangle\right] \\ b_j = \mathrm{Im}\left[\langle v_j|\Phi\rangle\right] \end{cases} \tag{2.29}$$

where $Sa = b$ gives the optimal Pauli strings. The matrix elements for S are the same, as the vectors $|v_j\rangle$ are identical in both cases. Since $|\Phi\rangle$ has a different form, the elements of b are modified and given by

$$b_j = \mathrm{Im}\left[\langle v_j|\Phi\rangle\right] = 2\tau \sum_{xy} p_x p_y \mathrm{Im}\left[\langle x|T^\dagger \sigma_j L_k T|y\rangle\langle y|T^\dagger L_k^\dagger T|x\rangle\right]. \tag{2.30}$$

Measuring Matrix Elements

To obtain the coefficients q_x and a_i, we need to measure various matrix elements. In general, we can decompose any operator into a sum over Pauli strings, $X = \sum_j x_i \sigma_i$. Since $\langle x|X|y\rangle = \sum_j x_i \langle x|\sigma_i|y\rangle$, we then need to measure $\langle x|\sigma_i|y\rangle$ for all Pauli strings σ_i. This can be done using the following identities:

$$2\mathrm{Re}\left[\langle x|X|y\rangle\right] = \frac{\langle x| + \langle y|}{\sqrt{2}} X \frac{|x\rangle + |y\rangle}{\sqrt{2}} - \frac{\langle x| - \langle y|}{\sqrt{2}} X \frac{|x\rangle - |y\rangle}{\sqrt{2}}, \tag{2.31}$$

$$2\mathrm{Im}\left[\langle x|X|y\rangle\right] = \frac{\langle x| + i\langle y|}{\sqrt{2}} X \frac{|x\rangle - i|y\rangle}{\sqrt{2}} - \frac{\langle x| - i\langle y|}{\sqrt{2}} X \frac{|x\rangle + i|y\rangle}{\sqrt{2}}. \tag{2.32}$$

In general, the state $(|x\rangle + i^p|y\rangle)/\sqrt{2}$, with $p \in \{0, 1, 2, 3\}$, requires a quantum circuit comprising m CNOT gates and having depth $m+1$, where m is the Hamming distance between the binary strings x, y [108]. Indeed, one can find an index k such that $x_k \neq y_k$. Without loss of generality, one can assume that $x_k = 1$ (otherwise, just invert the roles of x, y and replace p with $-p$ mod 4). One can then define the sets $S = \{l : x_l = 1, l \neq k\}$, $T = \{l : x_l \neq y_l, l \neq k\}$. Finally, starting from a register of n qubits prepared in $|0\rangle^{\otimes n}$, the desired state is obtained by: (i) applying a product of X gates on qubits in the set S, $\prod_{l\in S} X_l$, (ii) applying to qubit k the gate $g_p = H, SH, ZH, ZSH$, for $p = 0, 1, 2, 3$, respectively, and (iii) applying a product of CNOT gates to qubits in T controlled by qubit k, $\prod_{l\in S} c_k X_l$.

For local observables the state preparation is simpler, as described in the following. Consider a k-qubit observable $X^{(k)} \otimes \mathbb{I}_{n-k}$, with $X^{(k)}$ acting non-trivially on k qubits

out of a total of n qubits. Then

$$\langle x|X^{(k)} \otimes \mathbb{I}_{n-k}|y\rangle = \langle x_1, \ldots, x_k, x_{k+1}, \ldots, x_n|X^{(k)} \otimes \mathbb{I}_{n-k}|y_1, \ldots, y_k, y_{k+1}, \ldots, y_n\rangle \tag{2.33}$$

$$= \delta_{x_{k+1},y_{k+1}} \cdots \delta_{x_n,y_n} \langle x_1, \ldots, x_k|X^{(k)}|y_1, \ldots, y_k\rangle. \tag{2.34}$$

Thus we need only to prepare the states

$$\frac{|x_1, \ldots, x_k, x_{k+1}, \ldots, x_n\rangle + |y_1, \ldots, y_k, x_{k+1}, \ldots, x_n\rangle}{\sqrt{2}} = \frac{|x_1, \ldots, x_k\rangle + |y_1, \ldots, y_k\rangle}{\sqrt{2}} \otimes |x_{k+1}, \ldots, x_n\rangle. \tag{2.35}$$

Since k is typically small, only 1 or 2 qubits in most cases and independent of the system size, this state can be efficiently prepared. The form of Eq. (2.35) suggests a stochastic sampling method to determine which p_x's to store classically. For simplicity we describe the case of qubits in a line, and the indices $1, \ldots, n$ labelling the sites with the observable acting on the first k qubits. The general case is similar. Since the matrix elements will depend more heavily on qubits $1, \ldots, k+m$ for some cutoff m, we can sample with higher frequency on the first $k+m$ qubits and with lower frequency on the rest. In addition, in many cases we expect the dissipation channels L_k to reduce long range correlations, further increasing the convergence rate of local sampling.

Conservation of probability

The trace of the density operator, given by $\mathrm{Tr}\,(\rho) = \sum_x p_x = 1$, is preserved by time evolution generated by the Lindblad equation. Here we show that time evolution via Algorithm II also maintains the trace. The trace is preserved if the sum of all q_x's is zero at each time step. This requires summing the contributions to the q_x's from both the V_k and W_k terms as follows:

$$\begin{aligned}
\sum_y q_y &= \sum_y \left(-\tau p_y \mathrm{Re}\left[\langle y|T^\dagger L_k^\dagger L_k T|y\rangle\right] + \tau \sum_x p_x \left|\langle y|T^\dagger L_k T|x\rangle\right|^2 \right) \\
&= -\tau \sum_y p_y \langle y|T^\dagger L_k^\dagger L_k T|y\rangle + \tau \sum_x p_x \sum_y \langle x|T^\dagger L_k^\dagger T|y\rangle\langle y|T^\dagger L_k T|x\rangle \\
&= -\tau \mathrm{Tr}\left(\rho L_k^\dagger L_k\right) + \tau \sum_x p_x \langle x|T^\dagger L_k^\dagger L_k T|x\rangle \\
&= -\tau \mathrm{Tr}\left(\rho L_k^\dagger L_k\right) + \tau \mathrm{Tr}\left(\rho L_k^\dagger L_k\right) \\
&= 0.
\end{aligned} \tag{2.36}$$

Measuring observables

The result of the solving the QITE linear system at each time step, the weights $p_x(t)$ and the Hermitian QITE Hamiltonian $A(t)$, can be used to calculate the expectation value of any observable as a function of time. Inverting the Choi-Jamiołkowski isomorphism gives us the density operator $\rho(t) = \sum_x p_x(t)T(t)|x\rangle\langle x|T^\dagger(t)$. Observables O are then calculated as

$$\begin{aligned}\langle O(t)\rangle = \mathrm{Tr}\left(O\rho(t)\right) &= \sum_{xy} p_x(t)\langle y|OT(t)|x\rangle\langle x|T^\dagger(t)|y\rangle \\ &= \sum_{xy} p_x\langle x|T^\dagger|y\rangle\langle y|OT|x\rangle \\ &= \sum_{x} p_x\langle x|T^\dagger\left(\sum_{y}|y\rangle\langle y|\right)OT|x\rangle \\ &= \sum_{x} p_x(t)\langle x|T^\dagger(t)OT(t)|x\rangle.\end{aligned} \tag{2.37}$$

Beyond a certain number of qubits, storing all the $p_x(t)$'s is not possible, and a stochastic sampling approach is needed. Locality conditions suggest one possible approach to efficient sampling, described at the end of section (2.3), which converges faster than uniform random sampling. It is important to note that although the ansatz lies in a dilated Hilbert space, all measurements take place on the original system and no entangling operations between the system and ancilla are needed, and so no ancillae qubits are needed. In particular, for each time step measurements only on the original Hilbert space are used to determine the Hermitian matrix A. Expectation values of observables on this state are computed using the standard methods [65, 64].

The benefits of Algorithm II are that it requires no ancilla qubits, and no Hadamard test is required for measurements of observables. These characteristics, trading quantum for classical resources and simulating large quantum circuits using smaller quantum computers are important for near-term hardware [109, 110, 111, 112, 113]. In particular, Algorithm II allows for halving the number of required qubits as in Ref. [113], allowing simulation of larger physical systems by increasing the classical and quantum computational time while decreasing the required number of qubits. Its drawbacks are the sparse representation of the density matrix and the number of measurements required to evolve the system. We discuss this overhead in the following section.

2.4 Run time bounds, computational overheads, and errors

In this section, we discuss the run times, quantum and classical computational overheads, and errors associated with each algorithm. Other sources of errors, such as those associated with noisy hardware, are not addressed here as they are non-algorithmic errors.

Run time bounds

We first bound the run time of Algorithm I. For each time step in the Trotterization, the algorithm requires applying the imaginary time propagator $\exp(-H_2\tau)$, where $\tau = t/N$ and N is the number of Trotter steps for the time evolution. Assuming a local Lindblad equation, H_2 is a sum of m_2 local terms h_l such that $H_2 = \sum_{l=1}^{m_2} h_l$, where m_2 scales polynomially with system size. The imaginary time evolution operator $\exp(-H_2\tau)$ is implemented by additional Trotterization. For a given desired error ϵ_2, we Trotterize the imaginary time evolution into p_2 steps. From Eq. (3.8) of Ref. [114], we find that for $p_2 > 1/\epsilon_2$ the error in the p_2-step approximation is bounded by ϵ_2, assuming the number of Trotterization steps for time evolution N is sufficiently large that $3m_2tv_2/N < 1$, where $v_2 = \max_l\{\|h_l\|\}$.

Each term in the Trotterization is an imaginary time increment and so corresponds to a rotation by a unitary operator supported on D qubits where D is the domain size. An arbitrary D qubit unitary can be decomposed exactly into $O\left(D^2 4^D\right)$ single-qubit and CNOT gates [45]. The total contribution to running time from all the imaginary time evolutions is $O\left(Nm_2D^2 4^D/\epsilon_2\right)$.

Algorithm I also has additional unitaries $\exp(-iH_1\tau)$ interleaved between each QITE step, leading to an additional overhead. Because H_1 is a sum of local terms, $H_1 = \sum_{l=1}^{m_1} h_l$, $\exp(-iH_1\tau)$ needs to be Trotterized as well. Performing a similar analysis for the real time evolution, we find the total running time to be

$$T = O\left(Nm_1k^2 4^k/\epsilon_1 + Nm_2D^2 4^D/\epsilon_2\right), \tag{2.38}$$

where ϵ_1 is the allowable Trotter error for the real-time evolution and k is the maximum number of qubits acted on by each term in the Hamiltonian. In the first term on the right hand side, we have assumed that each $k-$local unitary can be exactly decomposed into $O\left(k^2 4^k\right)$ single qubit and CNOT gates [45].

A similar analysis can be carried out for Algorithm II, resulting in the same run-time up to constant factors with the following difference. The errors appearing in the run-time bound for Algorithm II do not include errors incurred from approximating

the density operator with a strict subset of all bit-strings. Although in principle any density operator can be represented by the sum $\sum_x p_x U|x\rangle \otimes \overline{U}|x\rangle$, this sum contains exponentially many terms and so only a subset of all possible bit strings can be included efficiently. Exclusion of bit-strings leads to an error in representing the state given by $\sum_{x\in I^c} p_x$, where I is an index set containing all bit strings to be included, and I^c is its complement. Whether a polynomial scaling of the number of included bit-strings is sufficient for a desired error will be problem dependent. In practice, this scaling can be assessed by stochastically sampling the bit-strings until the simulation converges.

Measurement and classical computational overheads

Provided that the finite domain approximation holds, the largest computational overhead (apart from running time) of both algorithms is the measurement overhead. For Algorithm I, this measurement overhead is the same as in the original QITE algorithm. State tomography over each domain consisting of D qubits needs to be carried out to construct the unitaries over that domain, requiring $O\left(4^D\right)$ measurements. Assuming a 1-dimensional lattice, there are $O\left(n/D\right)$ domains, and so the total measurement overhead is $O\left((n/D)4^D\right)$ per time step. Similar bounds can be obtained for lattices in higher dimensions.

Algorithm II requires measurement of the matrix elements $\langle x|U^\dagger \sigma_i U|y\rangle$ for all Pauli strings σ_i supported on a domain D (measured in qubits) and all bit strings in $x, y \in I$ for some subset I of the 2^n n-bit strings. Measuring all matrix elements necessitates running $O\left(L4^D|I|^2\right)$ circuits per time step, where L is the number of Lindblad operators on the domain and $|I|$ is the number of bit-strings included in the computation. For the algorithm to be efficient, the number of bit strings included in I must scale polynomially or slower with system size.

The finite-domain approximation required from QITE is accurate in many cases because the domain size D can generally be taken to be smaller for dissipative systems compared to the same system with no dissipation, as dissipation generally reduces a system's correlation length [115]. It should be noted that a reduced correlation length that decreases the cost for quantum algorithms might also permit an efficient classical description of the quantum evolution. This imprecise boundary is a consideration for quantum simulation algorithms generally and remains a topic of active investigation.

Algorithm	# of qubits	Circuits per Lindblad operator
I	$2n+1$	$(n/D)4^D$
II	n	$(n/D)4^D\|I\|^2$

Table 2.1: Asymptotic number of circuits required per time step per Lindblad operator for both algorithms for an open system on n sites. Here, D is the domain size, and I is is a subset of all n-bit strings for which the corresponding matrix elements are measured.

Table 2.1 summarizes the asymptotic scaling of the number of circuits required per time step of both algorithms for open quantum system dynamical simulation on n sites.

2.5 Quantum hardware demonstrations

We demonstrate both algorithms on IBM Quantum hardware for two cases: the spontaneous emission of a two level system (TLS) in a heat bath at zero temperature, and the dissipative transverse field Ising model (TFIM) on two sites. The TLS ($n = 1$ from Table 2.1) requires three physical qubits and one physical qubit to simulate with Algorithm I and II, respectively. The TFIM ($n = 2$ from Table 2.1) requires five and two physical qubits, respectively.

Considering Algorithm I, neither the TLS nor the two-site dissipative TFIM on 5 qubits have constant depth circuit decompositions; Trotterizing both the real and imaginary time propagators results in a circuit with depth linear in the number of time steps. The resulting circuits are too deep for near-term devices. To overcome this limitation, we recompile the circuits as in Ref. [100]. In all simulations, we correct for readout error using the built-in noise models in Qiskit [116, 56, 117, 55]. All measurements reported represent the average of 8192 shots and were repeated three times. Sampling noise in the measurement of the expectation value of the Pauli strings can lead to numerical instabilities in the QITE linear system. Therefore, when constructing the QITE matrix for Algorithm I, regularizers 1×10^{-6} and 0.01, for the TLS and TFIM, respectively, were added to the diagonal terms of the S matrix to increase the condition number of the matrix S following the procedure in Ref. [59]. No regularizers were used for Algorithm II.

Hardware demonstration of a two-level system

We first present results for the TLS model with the Hamiltonian

$$H = -\frac{\delta}{2}\sigma_z - \frac{\Omega}{2}\sigma_x \tag{2.39}$$

and the Lindblad operator $\sqrt{\gamma}\sigma_-$, where σ_- is the lowering operator, δ is the detuning, Ω is the Rabi frequency, and γ is the spontaneous emission rate. We consider here the overdamped case where γ is on the order of the other energies in in the system. It was found via numerical simulations that to accurately capture the dynamics only the Pauli strings in the set $\{\sigma_x \otimes \sigma_z, \sigma_y \otimes \sigma_x, \sigma_y \otimes \sigma_z, \sigma_z \otimes \sigma_x\}$ needed to be included in the QITE unitary.

We set $\delta = \Omega = \gamma = 1$, and the initial state was chosen to be the excited state. In Fig. 2.3, we show the populations of the ground and excited states, with the experimental data averaged from three runs on IBM's *ibmq_mumbai* [118] for Algorithm I and *ibmq_casablanca* [118] for Algorithm II. Good qualitative agreement is obtained for all observables, with the deviation between the theoretical and experimental curves largely due to gate errors as confirmed by numerical simulations and noisy hardware emulations.

We observe an initial exponential decay in the population of the excited state due to spontaneous emission into the bath followed by an approach to the non-equilibrium steady steady state (NESS) for $\gamma t \gg 1$. Damped Rabi oscillations are visible between these two regimes. The populations in the NESS can be interpreted as a balance between the spontaneous emission due to coupling to the bath and the absorption and stimulated emission due to the Hamiltonian driving term σ_x [121]. In the NESS, the combined spontaneous and stimulated emission rates are equal to the absorption rate.

In the absence of driving by an external electric field ($\Omega = 0$) the Hamiltonian is diagonal in the computational basis, resulting in the off-diagonal matrix elements $\rho_{01} = \overline{\rho_{10}}$ approaching zero as the system thermalizes. Figure 2.4 shows that these matrix elements remain non-zero as the NESS is approached, indicating that the hardware correctly obtains the expected quantum coherence as measured in the canonical basis. Also shown in Fig. 2.4 is the purity $\mathrm{Tr}\left(\rho^2\right)$, which does not correspond to a time-independent Hermitian observable on the system but can nonetheless be obtained from the density operator representation on the hardware. Time evolution preserves the inner product $\mathrm{Tr}\left(\rho^2\right) = \langle\rho|\rho\rangle$ on the quantum simulator, but the

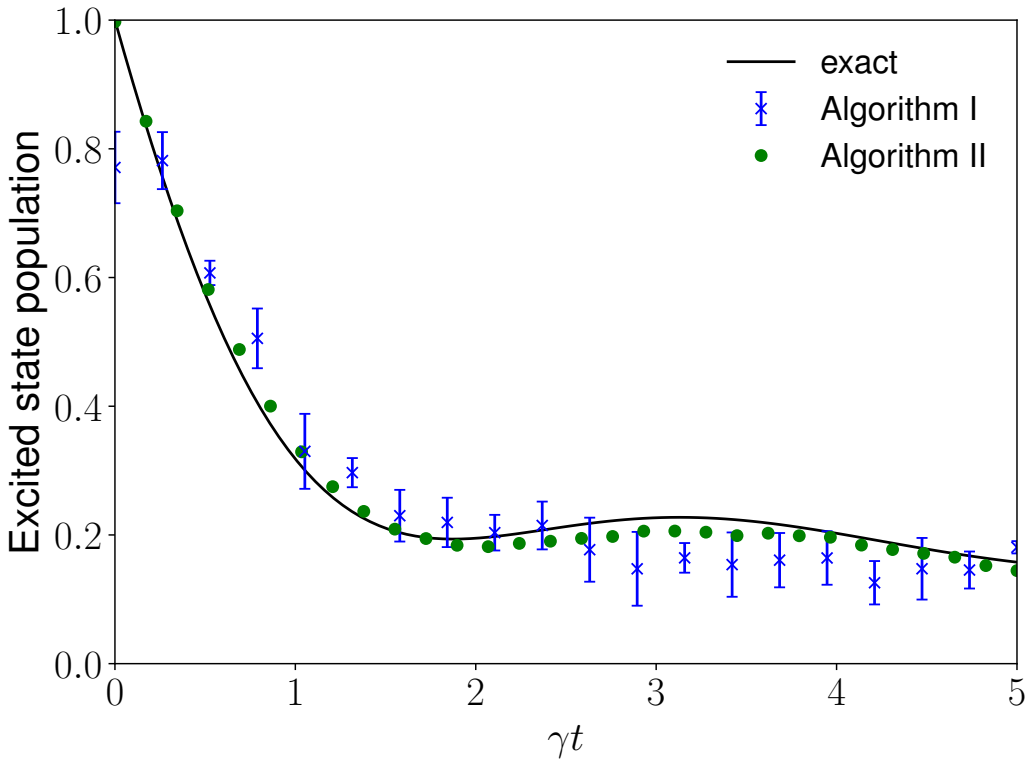

Figure 2.3: Population of the excited state from numerical simulations obtained in QuTiP [119, 120] (black line), hardware using Algorithm I on *ibmq_mumbai* [118] (blue crosses) and Algorithm II (green circles) on *ibmq_casablanca* [118]. The deviation between the theoretical and experimental curves is largely due to gate error. The system approaches a non-equilibrium steady state for $\gamma t \gtrsim 5$.

physical quantity, the normalized purity, $\mathrm{Tr}\left(\rho^2\right)/\mathrm{Tr}\left(\rho\right)$, is not constant.

The larger deviation between the hardware results and the exact results for Algorithm I is attributed to the fact that Algorithm I is a three-qubit circuit requiring two-qubit gates, which are generally of lower fidelity than single qubit gates. Since Algorithm I for the TLS is a single qubit circuit, there are no infidelity contributions from two-qubit gates. In addition, the circuits required for Algorithm I are deeper than for Algorithm II, resulting in more gate errors. An additional breakdown of the error contributions due to hardware error and algorithmic error is provided in Fig. 2.5, in which we compare the hardware results to noiseless numerical emulations. The noiseless numerical emulations were run with the same circuits as in the hardware trials but using IBM's *qasm_simulator*. From Fig. 2.5, we see that Algorithm I has a larger deviation between the emulation and hardware data. This difference can be accounted for by the fact that the hardware experiment for Algorithm II requires

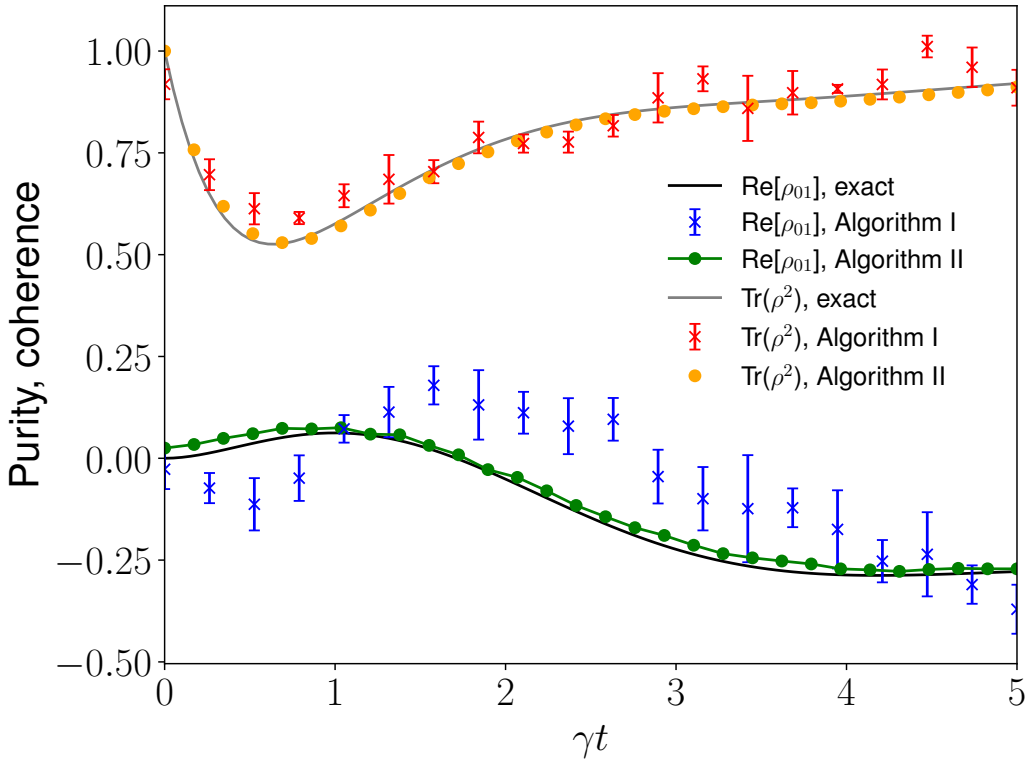

Figure 2.4: Purity, Tr $\left(\rho^2\right)$ (grey line) and off-diagonal term, Re $[\rho_{10}]$ (black line), corresponding to non-diagonal observables obtained in in QuTiP [119, 120]. Hardware results are shown for Algorithm I (purity, red crosses; Re $[\rho_{10}]$, blue crosses) and for Algorithm II (purity, orange circles; Re $[\rho_{10}]$, green circles). Hardware results for the observable Im $[\rho_{10}]$ agree with the exact solution similarly to Re $[\rho_{10}]$ but are omitted for clarity. For all hardware results for Algorithm I, the error bars are the standard deviation from three runs. The error bars for Algorithm II are smaller than the symbol size.

only a single qubit, so the density matrix for all time steps can be obtained from only single qubit rotations. Single qubit simulations can always be compiled to a constant depth regardless of the number of time steps, resulting in lower depth circuits and correspondingly lower total gate error. In addition, since 2-qubit gates are generally lower fidelity than single-qubit gates, there is no infidelity contribution due to 2-qubit gates in Algorithm II.

Since systems with larger or smaller dissipation rates correspond to different parameter regimes, we expect that the dissipation rate may also effect the accuracy of the algorithms. In general, we should expect larger algorithmic errors when larger dissipation rates are simulated since both the Trotter error and QITE error increase with the operator norm of the Lindblad operators. Larger dissipation rates corre-

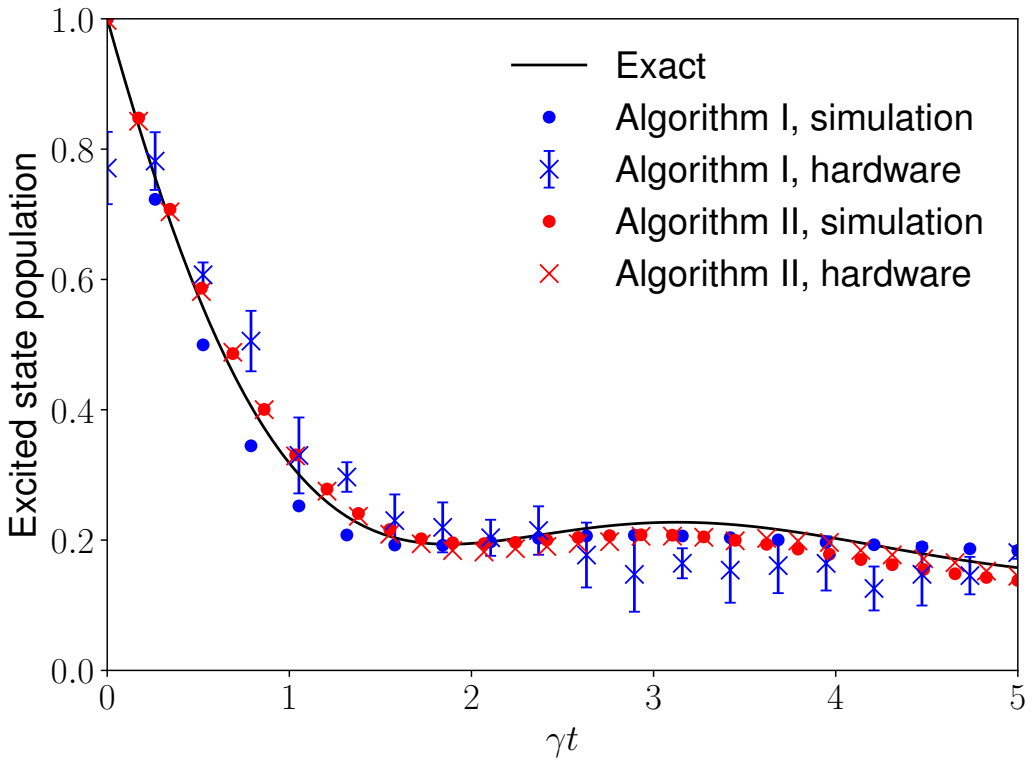

Figure 2.5: Excited state population for the two level system (TLS). The solid curve is the exact solution and the blue and red dots are noiseless numerical emulations of Algorithm I and II, respectively. The blue and red crosses are the hardware results presented in the main text for Algorithm I and II, respectively. The deviation between hardware and simulation results for Algorithm I are larger than for Algorithm II, which we attribute to hardware error resulting from the larger circuit depth and number of qubits needed for the Algorithm I.

spond to Lindblad operators with larger norms and hence larger algorithmic errors. To understand how increasing dissipation rates affect both algorithm's errors, we performed simulations of the 2-site TFIM with dissipation rates ranging from $\gamma = 0$ to $\gamma = 1$. Figures 2.6 and 2.7 shows the results of the simulations using Algorithm I and II, respectively. We see that in this specific case, which includes 16 Pauli strings for Algorithm I and all possible bit-strings for Algorithm II, Algorithm II performs qualitatively better than Algorithm I for all dissipation rates. Although Algorithm II performs better for the simulations shown in Fig. 2.7, we have not considered the error due to bit-string selection. For larger systems where all bit-strings cannot be included, there will be additional errors introduced by including only a strict-subset of all possible bit-strings.

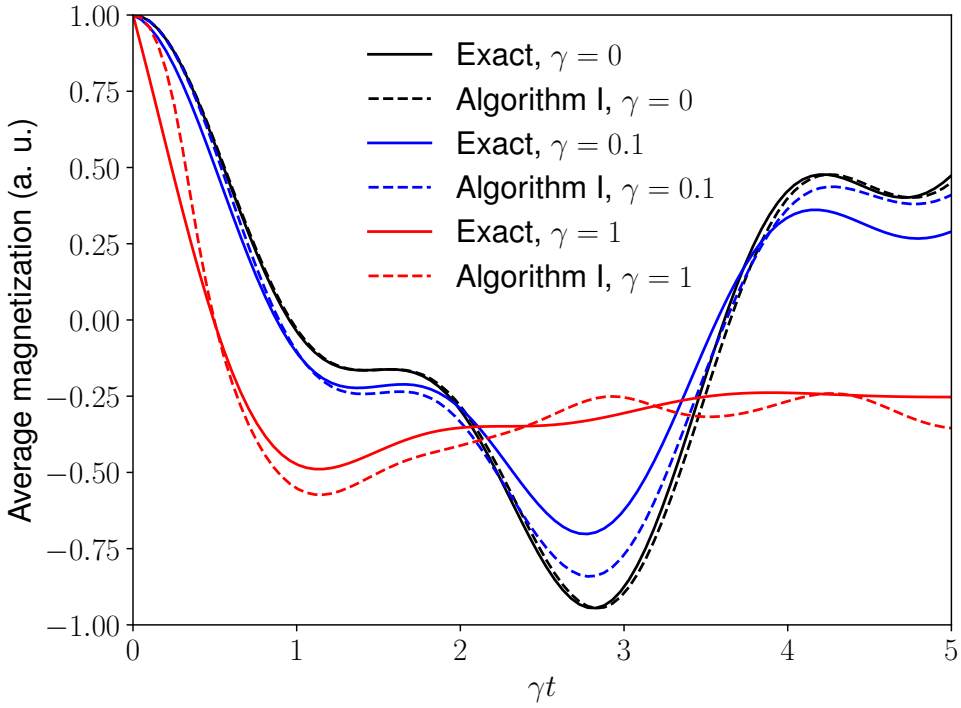

Figure 2.6: The effect of increasing the dissipation rate from $\gamma = 0$ to $\gamma = 1$. Noiseless simulation of Algorithm I using 16 Pauli strings. The same qualitative error is obtained for all dissipation rates simulated.

Hardware demonstration for a transverse field Ising model

We next present experimental and numerical results on the 2-site TFIM. The TFIM has the Hamiltonian

$$H = -J \sum_k \sigma_z^{(k)} \sigma_z^{(k+1)} - h \sum_k \sigma_x^{(k)} \tag{2.40}$$

and Lindblad operators $\sqrt{\gamma}\sigma_-^{(k)}$, with nearest neighbor coupling J, transverse magnetic field h, and decay rate γ. For this model, the number of required Pauli strings could not be reduced by symmetry in Algorithm I. To reduce circuit depth, 16 Pauli strings were randomly selected out of the 256 possible Pauli strings on 4 qubits to implement the QITE unitary. We chose 16 Pauli strings as a balance between too few Pauli strings, which results in a poor approximation to normalized imaginary time evolution, and too many Pauli strings, which results in a large computational overhead and an ill-conditioned QITE matrix.

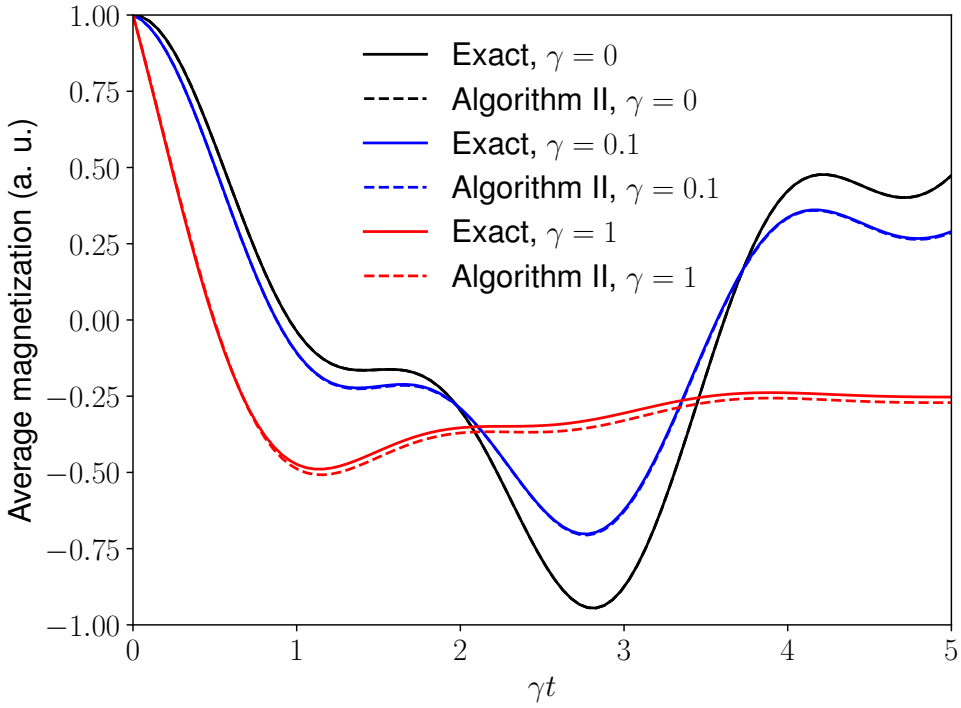

Figure 2.7: The effect of increasing the dissipation rate from $\gamma = 0$ to $\gamma = 1$. Noiseless simulation of Algorithm II.

Figure 2.8 shows the average magnetization of the dissipative TFIM with the initial state given by both spins in the spin up state and $J = h = 1$ and $\gamma = 0.1$. Oscillations in magnetization are evident due to the relatively large transverse field. We observe qualitative agreement between the theoretical and experimental curves from Algorithm I with a Trotter step $\gamma t/N \sim 0.5$. For the small system size of 2 sites, all 4 bit-strings on 2 qubit were included in Algorithm II. Experimental results for Algorithm II are also in good qualitative agreement with the exact curve for all times.

Exactly simulating the 2-site TFIM using Algorithm I requires measuring expectation values of the 256 Pauli strings on 4 qubits. To reduce the runtime of the Algorithm, we use a subset of all Pauli strings. We show in Fig. 2.9 that increasing the number of included Pauli strings beyond 16 has only a minor effect on the observables.

In Section 2.4, we discussed the runtime and resources required by both algorithms in a general setting. We now discuss the relative computational cost required

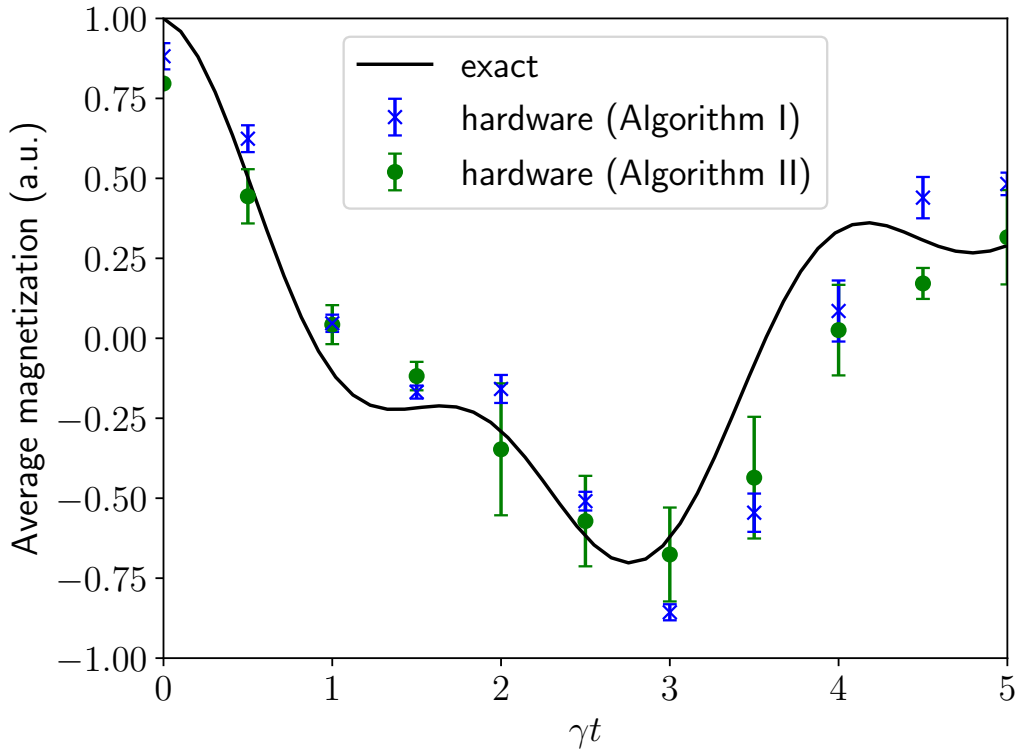

Figure 2.8: Average magnetization $N^{-1}\sum_i\langle Z_i\rangle$ for the dissipative transverse field Ising model on 2 sites (5 physical qubits for Algorithm I, 2 physical qubits for Algorithm II) using IBM Quantum's *ibmq_guadalupe* [118] for Algorithm I (blue symbols), and *ibmq_casablanca* [118] for Algorithm II (green symbols). Numerical solutions obtained in QuTiP are shown with black lines. The error bars for both algorithms are the standard deviation from 3 hardware runs. Both algorithms qualitatively agree with the exact dynamics for all simulated times. The deviation between the hardware results and the exact result for Algorithm II is due mainly to Trotter gate error.

by each algorithm for the specific case of the 2-site TFIM hardware simulations. For the simulations considered here, Algorithm I is able to accurately describe the dissipative dynamics when using 16 out of the total of 256 Pauli strings. Simulations of Algorithm I using up to 48 Pauli strings, shown in Fig. S2, show no significant increase in accuracy when using more than 16 Pauli strings. In general, the number of required Pauli strings will be problem dependent. Algorithm II requires measuring the matrix elements of all two-qubit Pauli strings at each time step, requiring 836 circuits per time step, versus only measuring expectation values of 16 operators in the case of Algorithm I, which requires 16 circuits. These measurements are only needed on a domain of fixed size.

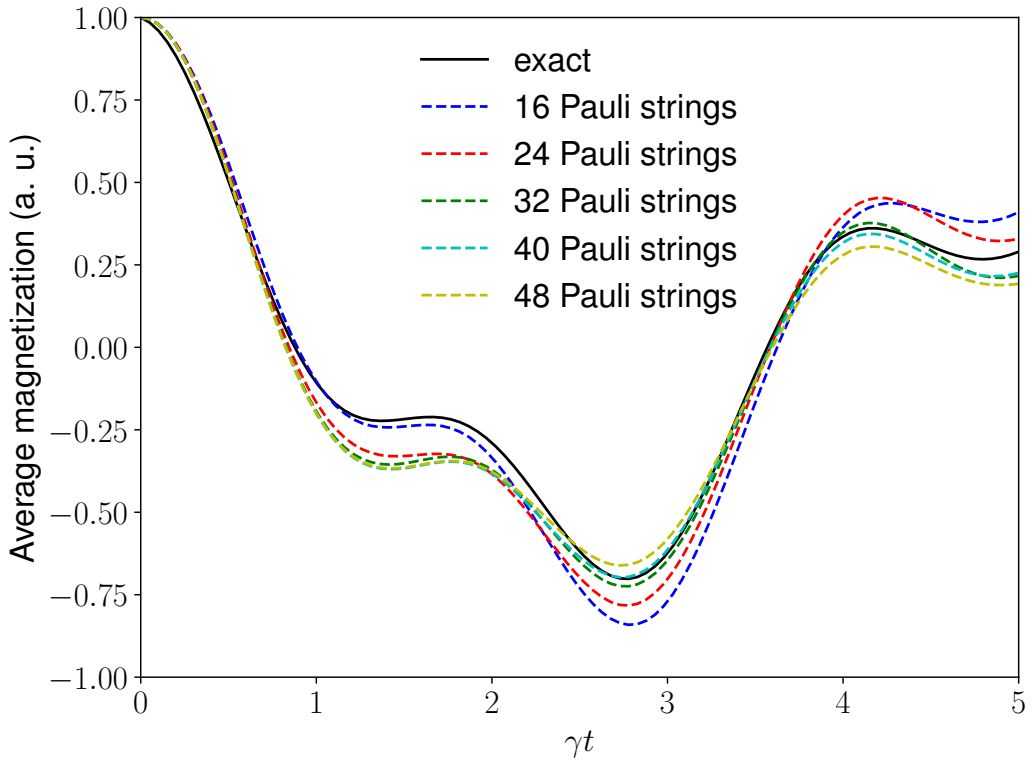

Figure 2.9: Noiseless numerical simulations for the transverse field Ising model (TFIM) using Algorithm I with increasing number of Pauli strings included. Here the dissipation rate is $\gamma = 0.1$. The black solid curve is the exact result, and the blue dashed curve is a simulation of Algorithm I using the same 16 Pauli's as in the main text. The red, green, light blue, and yellow dashed curves are noiseless numerical simulations of Algorithm I obtained from including an increasing number of Pauli strings in the simulation. From these simulations we see that only marginal increase in accuracy is obtained from including a larger number of Pauli strings.

For larger dissipation rates $\gamma \sim J, h$, separate numerical simulations, presented in Fig S3, show that both algorithms are able to accurately capture the magnetization dynamics. However, these simulations do not include the error incurred by including a subset of bit-strings in Algorithm II. The actual algorithmic error of Algorithm II will thus depend on the accuracy of the representation of the density matrix with a subset of bit-strings for the given problem. Stochastic sampling of bit-strings may be a viable approach for larger systems.

2.6 Summary

We have introduced digital quantum algorithms for the time evolution of open quantum systems described by a Lindblad equation based on quantum imaginary

time evolution. Algorithm I uses QITE to implement the non-unitary evolution introduced when the density operator is vectorized, whereas Algorithm II uses an adaptation of QITE to maintain a purification-based ansatz throughout the computation. Calculations for the spontaneous emission of a two level system and the dissipative transverse field Ising model, respectively, were carried out on IBM Quantum's quantum processors. Good qualitative agreement with the exact result was observed in all cases. These algorithms decrease the quantum resources required to simulate open quantum systems governed by Lindblad master equations on quantum hardware.

Chapter 3

SCALABLE CROSS-ENTROPY BENCHMARK OF MEASUREMENT-INDUCED PHASE TRANSITIONS ON A SUPERCONDUCTING QUANTUM PROCESSOR

[1] Hirsh Kamakari et al. Experimental demonstration of scalable cross-entropy benchmarking to detect measurement-induced phase transitions on a superconducting quantum processor. In: *arXiv preprint arXiv:2403.00938* (2024). URL: https://arxiv.org/abs/2403.00938.

In the previous chapter, we described two algorithms for simulating open quantum systems on a quantum computer. Although we were able demonstrate both algorithms on quantum hardware, due to the excessive circuit depth resulting from Trotterization we required fitting the circuits to a known ansatz. This approach is not scalable and was only used for demonstrations purposes. In general, Hamiltonian dynamics results in circuits which are too deep to run effectively on a quantum computer. Another class of quantum computations which exhibit interesting dynamics are random quantum circuits. These circuits can often be shorter depth than circuits for Hamiltonian dynamics, and therefore are a prime candidate for study on near term quantum computers. [122, 123]

In this chapter, we experimentally study measurement induced phase transitions(MIPTs) which result from random circuits consisting on unitary operations as well as measurements at random points in space-time. Although MIPTs have been previously studied on superconducting quantum processors [60], scalable demonstrations without any exponential overheads have not. Here, we present an experimental demonstration of MIPTs using a scalable cross-entropy benchmarking protocol [61].

3.1 Introduction

Quantum systems undergoing unitary evolution in the presence of an observer making measurements (monitored quantum systems) [124, 125, 126] exhibit unique dynamics, distinct from both thermalizing closed systems and conventional open quantum systems. When a system is weakly monitored and subject to sufficiently entangling unitaries, initial product states typically exhibit a linear in time growth of the entanglement entropy, before evolving into steady states where the entanglement

entropy admits a volume-law scaling [127, 128, 129, 130]. In contrast, strongly monitored systems are not able to support highly entangled states, resulting in area-law entanglement scaling even at long times. Separating the two phases lies a continuous phase transition, which was initially found theoretically in simplified quantum circuit models with mid-circuit measurements, and was later found to be generic to a wide range of monitored dynamics [131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145]. Such measurement-induced phase transitions (MIPTs) have recently garnered much interests [146], in part due to multiple theoretical viewpoints one can take in describing them [147, 148, 149, 150, 151].

An experimental observation of MIPTs was recently demonstrated on IBM quantum hardware with up to 14 qubits [60]. By directly measuring the entanglement entropy after a comprehensive quantum tomography of the steady states, Koh et al. [60] were able to observe an MIPT and confirm the competing effects of random unitaries and mid-circuit measurements. However, the experiment required over 5200 qubit-hours and is limited in scalability due to the exponential cost of quantum state tomography and post-selection of measurement outcomes. The lifetime of superconducting qubits also puts a stringent limit on the circuit depth (as well as on system size when circuit depth scales with the number of qubits), since mid-circuit measurements can be an order of magnitude slower than two-qubit unitary gates. To avoid mid-circuit measurements, a space-time duality mapping was introduced [152, 153] and recently implemented on Google's superconducting processor [154], where MIPT-like physics was observed in 1D unitary circuits with a reduced number of post-selections, and at the boundary of shallow 2D unitary circuits of 70 qubits without post-selection. Alternatively, order parameters based on reference qubits can be used to efficiently and scalably probe MIPTs [155], where post-selection can be avoided with an accompanying classical simulation. The use of a reference qubit to probe MIPTs has been demonstrated in trapped ion systems for Clifford circuits [156], featuring a high gate fidelity and non-local qubit connectivity. However, critical exponents at the transition were not obtained from previous experimental data.

In this chapter, we explore experimental realizations and efficient probes of MIPTs in prototypical hybrid Clifford circuit models with up to 22 physical qubits in less than 8 qubit-hours on IBM superconducting devices. By combining a recently proposed cross entropy benchmark (XEB) protocol [61] with a Clifford-based circuit compression [62], we circumvent the exponential overhead in post-selection and

tomography, and manage to probe MIPTs in systems with larger effective system size than the available number of physical qubits while minimizing the effect of noise. Moreover, the circuit compression allows us to investigate circuit models with all-to-all connectivity on IBM's 2D layout, due to reduced non-locality in compressed circuits. We obtain precise estimates of critical exponents that are comparable to theoretical predictions. Comparing the experimental data to numerical simulations with stochastic erasure error, we give rough estimates of effective noise rates per gate of the superconducting devices, which are comparable to numbers IBM reported. Qualitative differences between experimental data and numerical simulations are also observed, which we understand to be macroscopic manifestations of real device errors that are beyond our simple noise model.

This work paves the way for scalable studies of other critical phenomena on near-term quantum hardware and provides a benchmarking tool for quantum circuits with mid-circuit measurements.

3.2 Circuit model and cross entropy benchmark

We consider a family of random circuits, where each circuit consists of two stages: an purely unitary "encoding stage" consisting of t_{encoding} layers, and a "bulk stage" consisting of t_{bulk} layers with both unitary gates and mid-circuit measurements, see Fig. 3.1(a). For an L-qubit circuit, both stages must contain a number of layers scaling at least linearly with L for the system to enter a steady state, particularly when the steady state has volume-law scaling of entanglement entropy.

The protocol involves the application of the same circuit to two different initial states, ρ and σ, and a comparison between the two ensembles of measurement records. For a given circuit C with N mid-circuit measurements, a measurement record $\mathbf{m} = (m_1, m_2, \ldots, m_N)$, where m_j are the outcomes (0 or 1) from each mid-circuit measurement, is sampled by running the circuit on quantum hardware with input state ρ. The sampled measurement records obey a probability distribution, which we denote as $p^{\rho}_{\mathbf{m}}$. We also implement the sampling experiment using classical simulations of the same circuit, but with a different (stabilizer) initial state σ. The corresponding measurement record probabilities is similarly denoted as $p^{\sigma}_{\mathbf{m}}$. The cross entropy acts as a distance measure between the two distributions, and is defined for this circuit as

$$\chi_C = \frac{\sum_{\mathbf{m}} p^{\rho}_{\mathbf{m}} p^{\sigma}_{\mathbf{m}}}{\sum_{\mathbf{m}} (p^{\sigma}_{\mathbf{m}})^2}, \tag{3.1}$$

which can be estimated by taking the sample average of $p^{\sigma}_{\mathbf{m}} / \left(\sum_{\mathbf{m}} (p^{\sigma}_{\mathbf{m}})^2\right)$ over many

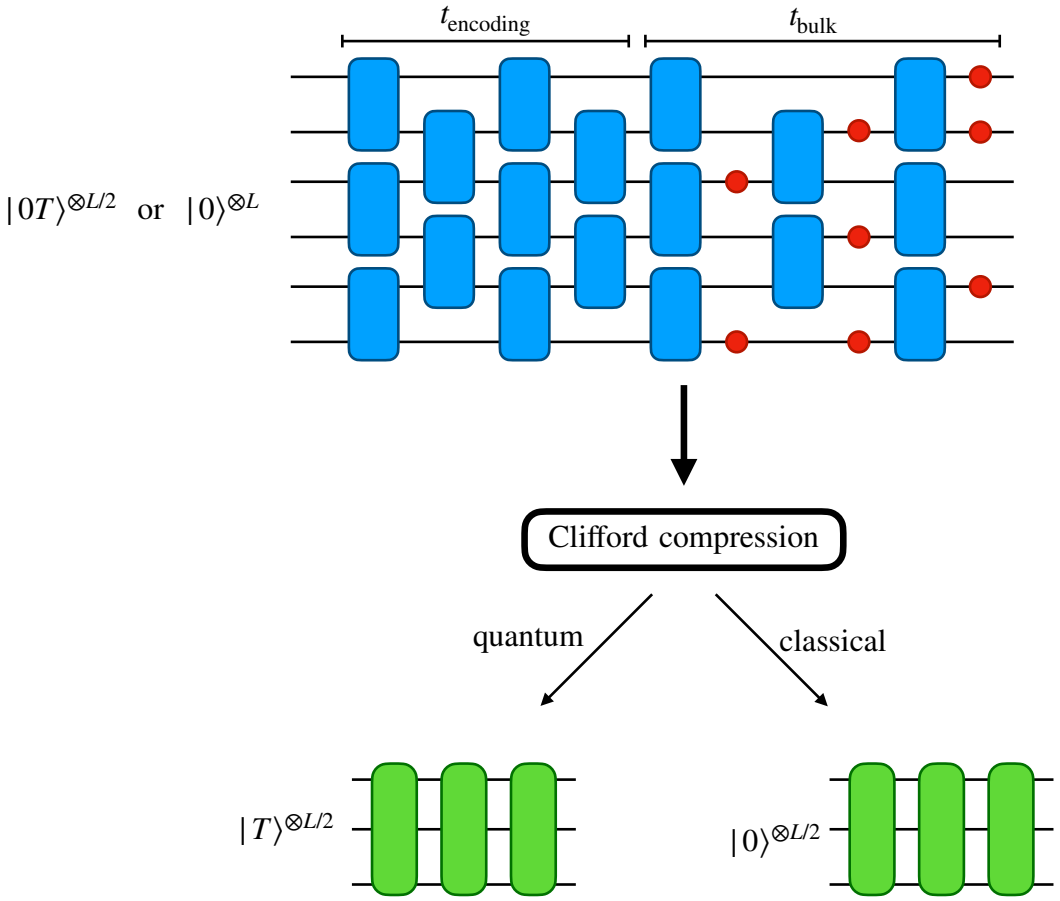

Figure 3.1: Schematic of the protocol demonstrated in this chapter. (a) We construct an L-qubit Clifford circuit consisting of t_{encoding} encoding layers, and t_{bulk} bulk layers. Each layer (shown in the red dashed box) consists of L random 2-qubit Clifford unitaries and each bulk layer additionally contains Pauli-Z measurements . The measurements are performed at each spacetime location of the bulk independently and with probability p. We choose the initial state $|\phi\rangle$ to be either $|0T\rangle^{\otimes L/2}$, where $|T\rangle$ is a magic state, or $|0\rangle^{\otimes L}$. (b) After circuit compression, we obtain an $L/2$ qubit circuits consisting of at most $L/2$ multi-qubit Pauli measurements which may not be geometrically local and have to be compiled into nearest-neighbor two-qubit gates and single-site measurements. The compressed initial state is $|\phi'\rangle = |T\rangle^{\otimes L/2}$ or $|0\rangle^{\otimes L/2}$. Only circuits with initial magic states are run on hardware.

runs of the quantum circuit with input state ρ. We then average over random circuits C, obtaining the final cross entropy for a given measurement rate as $\chi = \mathbb{E}_C \chi_C$. As shown in [61], for $\rho \neq \sigma$ and in the absence of noise, the quantity χ acts as an order parameter which, in the thermodynamic limit, approaches 1 when the system is in the volume-law phase (small p) and approaches a constant strictly less than 1 in the area-law phase (large p). Intuitively, χ measures the distinguishability of the two initial states by comparing mid-circuit measurement records, after the two initial states are "scrambled" by the encoding unitary. Previously, the linear cross entropy has been used as a figure of merit for random circuit sampling [157, 123, 158, 159]. For χ to be efficiently obtainable from quantum and classical hardware, the probabilities $p^\sigma_{\mathbf{m}}$, as well as $\sum_{\mathbf{m}} (p^\sigma_{\mathbf{m}})^2$, need to be efficiently classically computable. This is possible when the bulk of the circuit contains only Clifford operations and when the input state σ is a stabilizer state. The cross entropy protocol is similar in spirit to hybrid quantum-classical observables used in previous experiments [154, 156] (see also [160, 161]) and, as we will show, allows us to probe critical properties on noisy processors without error mitigation.

We implemented this approach on IBM Quantum processors. The systems we considered are a 1D chain with nearest-neighbor qubit connectivity and an infinite-dimensional system with all-to-all qubit connectivity. We chose the initial L-qubit states on the quantum processor in both cases to be $\rho = |0T0T \cdots 0T\rangle\langle T0T0 \cdots T0|$ with $|T\rangle = (|0\rangle + \exp(i\pi/4)|1\rangle)/\sqrt{2}$, the alternating magic state, and $\sigma = |0^{\otimes L}\rangle\langle 0^{\otimes L}|$, the all-zero state. We note that ρ is not a stabilizer state. For the alternating magic states, the number of T gates grows linearly with the number of qubits, so that an exact simulation of the circuit is classically intractable [162, 163, 164, 165]. For all experiments, the circuits are constructed using alternating layers of unitaries and measurements. Each unitary layer consists of $L/2$ two-qubit unitary gates, sampled uniformly from the two-qubit Clifford group. For the 1D chain, the two-qubit Clifford gates are applied in a brickwork pattern on nearest-neighbor qubits. For the infinite dimensional system, $L/2$ two-qubit unitaries are applied to pairs of qubits selected uniformly at random. Each measurement layer, in both the 1D chain and the infinite-dimensional system, consists of single-qubit Z measurements occurring on each qubit with probability p. For both systems, we used an encoding ratio and bulk ratio of 3, namely $t_{\text{bulk}} = t_{\text{encoding}} = 3L$. For all experiments, we generated 1000 random circuits for each (L, p) pair, and each circuit was run 1000 times on the *ibm_sherbrooke* machine.

3.3 Compression of Clifford circuits with magic initial states

The resulting circuits with the above properties have at most L mid-circuit measurements per qubit, which are relatively slow operations and introduce both readout and quantum state errors, and so they cannot be executed while preserving adequate fidelity. We therefore employ a circuit compression scheme which exploits the input state being an alternating magic state and the circuit bulk being fully Clifford. After circuit compression, we obtain a circuit with $L/2$ hardware qubits and at most $L/2$ multi-qubit measurements, significantly fewer than an average of $pt_{\text{bulk}}L^2$ measurements in the original uncompressed circuits. For each measurement record of the compressed circuit, we can further use it as a seed to classically sample a measurement record according to the ensemble of the uncompressed circuit. The initial state of the circuit is now $|T\rangle^{\otimes L/2}$, see Fig. 3.1(b). The multi-qubit measurements are generally not geometrically local, and must be compiled into local gates and local measurements (which are now mid-circuit measurements). As a result, the circuit compression reduces the number of two-qubit gates from $3L^2$ to $L^2/2$, but increases the circuit depth from $9L$ to an average of L^2, see Sec. 3.3 for additional details. All circuits used in our experiments use Clifford compression, bringing larger system sizes within the range of accessibility.

Here we describe the Clifford based compression algorithm we use to reduce the required number of physical qubits by a factor of two, as well as to reduce the total number of mid-circuit measurements to equal the number of physical qubits. The compression is based on Ref. [62] with an improvement that removes the requirement for dynamic circuits (adaptivity), but instead using an efficient classical simulation and classical coin flipping. Here we first summarize the compression algorithm stated in Ref. [62], and then explain how to remove the adaptivity.

Summary of the compression algorithm

In a particular circuit realization the unitaries and the measurements can be written as

$$C_{\mathbf{m}} = \ldots U_3 M_{m_2} U_2 M_{m_1} U_1 . \tag{3.2}$$

Here m_j is the j-th measurement outcome of the entire record, and correspondingly $M_{m_j} = (1 + (-1)^{m_j} P_j)/2$ is the j-th projection operator, with P_j the Pauli operator being measured. Moving all unitaries past the measurements to the right, we can equivalently write

$$C_{\mathbf{m}} = \ldots \widetilde{M}_{m_2} \widetilde{M}_{m_1}, \tag{3.3}$$

where

$$\widetilde{M}_{m_j} = \frac{1}{2}(1 + z_j \widetilde{P_j}), \quad \widetilde{P}_j = U_1^\dagger U_2^\dagger \dots U_j^\dagger P_j U_j U_{j-1} \dots U_1 \tag{3.4}$$

are now multi-site Pauli measurements and $z_j = (-1)^{m_j}$.

Let $A = \{1, \dots, k\}$, and $B = \{k+1, \dots, N\}$. Following Ref. [62], we state without proof that the following algorithm correctly samples an output bitstring of the circuit C on a input state in the new basis, with input states of the form $|\psi\rangle = |\phi_A\rangle \otimes |0_B^{\otimes N-k}\rangle$.

1. Initialize the quantum state $|\phi_A\rangle$, define the initial stabilizer group $\mathcal{S} = \langle Z_{k+1}, \dots Z_N \rangle$, and let the Pauli operators be $\{\widetilde{P}_j\}$.

2. Consider each $\widetilde{P}_j$ in increasing order of j. For each j there are three possible cases:

   a) $\widetilde{P}_j \in \mathcal{S}$. In this case the measurement result is deterministic, and can be classically computed and we do not need to update the state or $\mathcal{S}$.

   b) $\widetilde{P}_j \notin \mathcal{S}$, and it anticommutes with at least one element $Q \in \mathcal{S}$. In this case, the measurement result of $\widetilde{P}_j$ is equally likely $z_j = \pm 1$. We can flip a classical coin to sample z_j. Further, we need to account for the change in the state, which can be shown to be

   $$|\phi\rangle \to V_j(z_j)|\psi\rangle \tag{3.5}$$

   where $V_j(z_j)$ is a Clifford unitary operator

   $$V_j(z_j) = \frac{1}{\sqrt{2}}(Q + z_j \widetilde{P}_j). \tag{3.6}$$

   Instead of evolving the state and updating $\mathcal{S}$, we adopt the Heisenberg picture and modify all subsequent measurements $\widetilde{P}_{k>j}$ as follows,

   $$\widetilde{P}_k \to V_j(z_j)^\dagger \widetilde{P}_k V_j(z_j), \quad \forall k > j. \tag{3.7}$$

   c) $\widetilde{P}_j \notin \mathcal{S}$, and it commutes with all elements of $\mathcal{S}$. It then necessarily commutes with $Z_{k+1}, \dots, Z_N$ since these stabilizers are permanent, as we can check at the end of the algorithm (see comment 2 below). It follows that $\widetilde{P}_j$ only contains the identity operator or the Pauli Z operator on B. We can then consider a truncated Pauli operator that is supported only on A,

   $$\widetilde{P}_j^A \coloneqq \eta_j \cdot \widetilde{P}_j|_A, \tag{3.8}$$

where $\widetilde{P}_j|_A$ is the restriction of $\widetilde{P}_j$ on A, and the sign $\eta_j = \pm 1$ can be chosen such that for any state $|\phi_A\rangle$ we have

$$\langle\phi_A|\widetilde{P}_j^A|\phi_A\rangle = \langle\phi_A \otimes 0_B^{\otimes N-k}|\widetilde{P}_j|\phi_A \otimes 0_B^{\otimes N-k}\rangle. \tag{3.9}$$

The measurement of $\widetilde{P}_j$ on the joint system AB can therefore be faithfully simulated by a measurement of $\widetilde{P}_j^A$ on just A. We perform this measurement on the state $|\phi_A\rangle$, update the state accordingly and record the measurement result z'_j. We then update the stabilizer group as

$$\mathcal{S} \to \langle \mathcal{S}, z'_j \widetilde{P}_j^B \rangle. \tag{3.10}$$

We see that in this algorithm

1. Cases (1) and (2) can be accounted for by classical simulation, and only in case (3) a quantum operation on $|\phi_A\rangle$ needs to be performed.

2. The stabilizer group $\mathcal{S}$ gets augmented only in case (3), and can be augmented at most k times. Once an operator is added into $\mathcal{S}$, it will remain in $\mathcal{S}$ until the algorithm terminates.

In this way, a given sequence of multi-site measurements can be simulated by a "compressed circuit" with at most k multi-site measurements on A, as well as classical coin flips, up to a polynomial time overhead.

Removal of adaptivity

A major problem of the above algorithm is that the update of the stabilizer group $\mathcal{S}$ in case (c) depends on the quantum measurement result z'_j. Not knowing z'_j before the circuit execution will lead to the lack of knowledge of the sign of $Q \in \mathcal{S}$ in case (b) if occuring after the update of $\mathcal{S}$ due to case (c). Here we show the adaptivity can be removed by proving that the effect of flipping signs of z'_j or Q can be captured by classical postprocessing.

In order to prove it, we first notice that $Q \to -Q$ is equivalent to $z_j \to -z_j$ in Eq. 3.6 ($V \to -V$ has no effect on Eq. 3.7). We additionally notice that

$$V_j(-z_j) = QV_j(z_j)Q, \tag{3.11}$$

so that for any $k > j$,

$$\begin{aligned} V_j(-z_j)^\dagger \widetilde{P}_k V_j(-z_j) &= QV_j(z_j)^\dagger Q\widetilde{P}_k QV_j(z_j)Q \\ &= \lambda_{Q,\widetilde{P}_k} QV_j(z_j)^\dagger \widetilde{P}_k V_j(z_j)Q \\ &= \lambda_{Q,\widetilde{P}_k}\lambda_{Q,V_j(z_j)^\dagger \widetilde{P}_k V_j(z_j)} V_j(z_j)^\dagger \widetilde{P}_k V_j(z_j), \end{aligned} \tag{3.12}$$

where we have defined the commutator of Pauli operators A, B

$$AB = \lambda_{A,B} BA. \tag{3.13}$$

Eq. (3.12) implies that flipping measurement results z'_j at most result in sign changes of the subsequent measurements operators $\widetilde{P}_{k>j}$, and such sign dependence can be classically captured. In practice, we can first determine the form of each Pauli operator to be measured on A in the compressed circuit, and assume they all have +1 sign; the adativity can be re-introduced in post-processing, by flipping the measurement results appropriately.

Decomposition of the Pauli-based computing model to a common gate set

Here we describe an algorithm to decompose each multi-qubit Pauli measurement in Eq. (3.3) to

$$P_j = \left(\prod_i C_i\right)^\dagger Z[k] \left(\prod_i C_i\right),$$

where $\{C_i\}$ contains up to m single-qubit Clifford operations and $2m$ CNOT gates. For a Pauli string $P_j = \otimes_{i=1}^m P_j[i]$, where $P_j[i] \in I, X, Y, Z$, we first convert each X and Y to a Pauli Z at qubit i by a single-qubit Clifford operation $C[i]$, i.e., $C[i]P_j[i]C[i]^\dagger = Z[i]$. After this step, the Pauli string becomes a string of Is and Zs. We note the fact that $\mathrm{CNOT}_{1,2}(I \otimes Z)\mathrm{CNOT}_{1,2} = (Z \otimes Z)$ and $\mathrm{CNOT}_{1,2}(Z \otimes Z)\mathrm{CNOT}_{1,2} = (I \otimes Z)$. Thus we first sequentially convert the Pauli string to the form of $I...IZ...ZI...I$ by converting adjacent ZI or IZ to ZZ, and then sequentially convert it to $I...IZI...I$ with a single Z in the middle by converting adjacent ZZ to IZ or ZI.

By using the above algorithm for the decomposition of $P_{j=1}$, we obtain $P_{j=1} = (\prod C_i)^\dagger Z[k](\prod C_i)$. However, instead of naively applying the algorithm for each P_j, we first "absorb" $(\prod C_i)^\dagger$ into the rest of the Pauli strings by $P_j \to (\prod C_i)P_j(\prod C_i)^\dagger$, and then apply the above algorithm to the next Pauli measurement. By doing such "absorption," we roughly reduce the number of CNOT gates by half. Finally, the compressed circuit is decomposed to at most m^2 single-qubit gates and $2m^2$ CNOT gates.

Resource reduction after circuit compression

	Before compression	After compression
Num. hardware qubits	L	$L/2$
Average depth	$9L$	L^2
Avg. num. 2 qubit gates	$3L^2$	$L^2/2$
Avg. num. measurements	$3L^2p$	$L/2$

Table 3.1: Hardware resources required before and after Clifford circuit compression for a fixed L and p. The number of hardware qubits, average depth, and average number of 2 qubit gates required are reduced by a constant factor after compression, whereas the average number of measurements is reduced by a factor of L and is independent of p. The values in this table apply both to the 1D system as well as the all-to-all system.

In Table 3.1 we present a summary of the quantum hardware resource requirements before and after circuit compression. Here we are setting $t_{\text{bulk}} = t_{\text{encoding}} = 3L$ and using an initial ρ state that is an alternating magic state.

3.4 Qubit selection

For the 1D-chain experiments, qubits were selected heuristically at submission time based on the one-qubit gate, two-qubit gate, and readout error rates provided by IBM in their hardware calibration data. We selected the qubits based on a minimization of the average errors that would occur in all circuits based on the number and placement of gates and measurements in the circuits. For the 1D-chain experiment with $\rho \neq \sigma$, the qubits were selected heuristically at run time as in Ref. [60]. The qubits we selected based on the one and two qubit gate error rates, ϵ^{1q} and ϵ^{2q}, respectively, as well as the qubit readout error ϵ^{ro} rates provided by IBM in their hardware calibration data. Denoting by χ the set of qubit selections which contain all $L/2$ qubits in a connected chain, an average circuit error for circuit C is calculated as

$$\mathcal{E}_{x\in\chi}[C] = \sum_{j\in x}(\epsilon^{1q}N_j^{1q}[C] + \epsilon^{2q}N_j^{2q}[C] + \epsilon^{ro}N_j^{ro}[C]), \tag{3.14}$$

where the subscript j represents the j'th qubit in the qubit set x, and the function $N_j^{1q,2q,ro}$ computes the number of single qubit gates, two qubit gates, and measurements, respectively, acting on qubit j in the circuit C. The qubit chain used in the experiment is then selected as the one which minimizes the average error over all circuits C, $\text{argmin}_{x\in\chi}\mathbb{E}_{C\in\mathcal{C}}\mathcal{E}_x[C]$.

For the all-to-all and $\rho = \sigma$ experiments, we used the same qubit layouts that were selected for the 1D-chain.

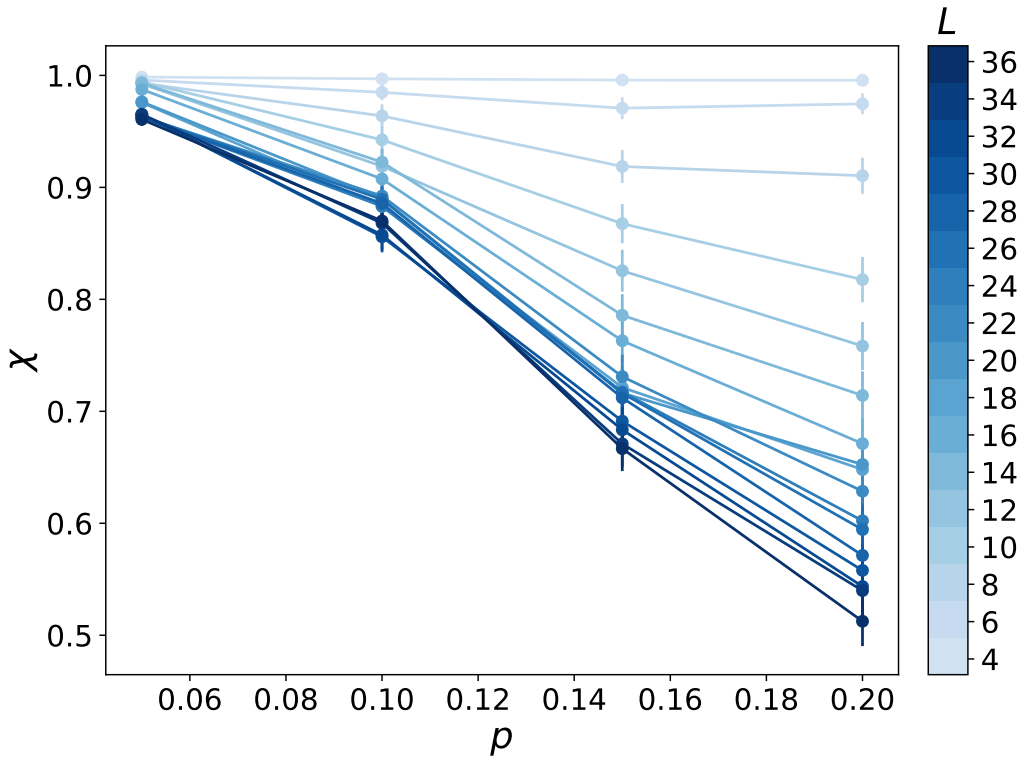

Figure 3.2: Cross entropy for identical initial states ($\rho = \sigma$) obtained from *ibm_sherbrooke* with up to 18 physical qubits (equivalent to a system size of $L = 36$ qubits before compression). The initial state for both ρ and σ is chosen to be the all-zeros state, with 2-qubit gates acting on nearest neighbors before compression. The errors incurred from the physical qubits results in a cross entropy lower than the theoretical value of 1. Larger systems have more measurements and more gates, incurring a larger overall error in the cross entropy.

3.5 Results for $\rho = \sigma$

We first present the experimental results when we set $\rho = \sigma$ to provide a benchmark of the hardware performance. We obtain the circuits from the compressed 1D circuits, but replace all $|T\rangle$ states with $|0\rangle$ states, so that the initial states on both ρ and σ are the all-zero state. In this case, since the circuits run on both the quantum and classical sides are identical, we expect to observe $\chi = 1$ for all L and for all p in the absence of any noise or hardware errors. The deviation of the cross entropy from 1 therefore provides a measure of the overall errors and noise in the circuit, which could be due to various sources such as gate errors, qubit decay and dephasing, and cross-talk from mid-circuit measurements.

The results are shown in Figure 3.2, where we observe that χ decreases when either L or p increases. For increasing L at fixed p, we have more mid-circuit measurements in both uncompressed and compressed circuits. For increasing values of p at fixed L, we do not necessarily have more measurements in the compressed circuit (they are upper bounded by L), but the measurement results will be more strongly correlated, see Section 3.3. Our noisy simulations in Section 3.12 of uncompressed circuits are qualitatively consistent with Fig. 3.2.

3.6 Results for 1D chain, $\rho \neq \sigma$

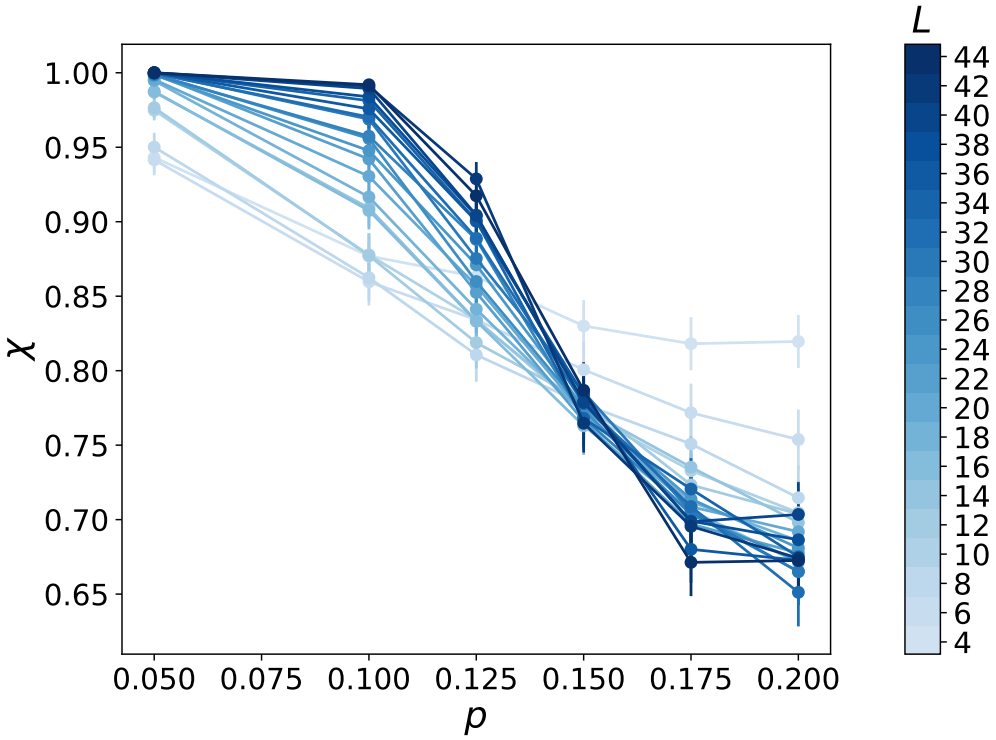

Figure 3.3: Cross entropy χ for 1D chains with up to 22 physical qubits (corresponding to a system size of $L = 44$ qubits before compression) computed on *ibm_sherbrooke*.

We next present experimental results for the 1D chain for $\rho \neq \sigma$. Fig. 3.3 shows the cross entropy curves obtained from the 127 qubit $ibm\_sherbooke$ device. Qualitatively, we see the expected characteristics as described below Eq. (3.1), namely with increasing L the approach of χ to 1 for smaller values of p, and the saturation of χ to a constant < 1 for larger values of p. The curves for different L cross at the critical point with $p_c \in (0.15, 0.175)$.

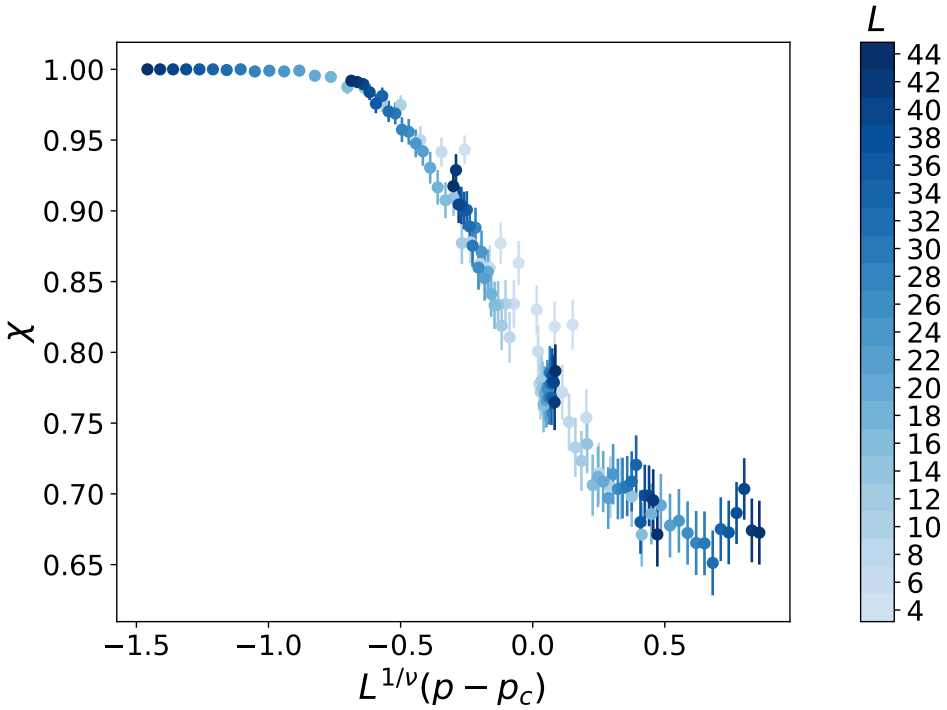

Figure 3.4: Collapse of cross entropy curves near the critical point obtained by minimizing the scatter of all points to an unknown scaling function. The fitting procedure gives a critical measurement rate of $p_c = 0.14 \pm 0.01$ and critical exponent $\nu = 1.4 \pm 0.5$.

The cross entropy is related to a domain wall free energy in an associated statistical mechanics model [61], and whose value near the critical point depends only on the ratio of the system size and the correlation length, according to standard scaling hypotheses. We verify this hypothesis by collapsing the data from different system sizes L and measurement probabilities p to an unknown but universal scaling function F:

$$\chi(L, p) = F\left[L^{1/\nu}(p - p_c)\right], \tag{3.15}$$

where ν is the critical exponent that controls the divergence of the correlation length, and p_c is the critical measurement rate [166]. With F unknown, we follow standard methods [60, 167] to choose the parameters p_c and ν so as to optimize the quality of the data collapse, see Sec. 3.8 for details. Such a procedure allows us to extract best fits for the critical measurement rate $p_c = 0.14 \pm 0.01$ and for the critical exponent $\nu = 1.4 \pm 0.5$ at the 90% confidence level, see results in Figure 3.4. Our reported values of p_c and ν are consistent with classical numerical calculations in the presence of 0.1% erasure noise, see Sec. 3.12.

For chains of fewer than 10 qubits, we see a stronger finite-size effect as indicated by deviating cross entropy curves as well as the saturation to a larger final value for large p (see Figure 3.4). Removing the smaller system sizes from the fitting allows for the critical values to be obtained while reducing finite-size effects. Although we extract the same value of ν whether we remove the smaller systems or not, the resulting data collapse has a lower variance, with all data points lying closer to a single curve. Including only the larger systems leads to fitting critical values of $\nu = 1.4 \pm 0.4$ and $p_c = 0.156 \pm 0.009$, see Figure 3.5. The value of ν when we fit all the data is $\nu = 1.4 \pm 0.3$ and the value of p_c is $p_c = 0.144 \pm 0.007$, and so are qualitatively the same within error bars. Although we obtain the same fit values, removing the small system sizes leads to a data collapse with reduced broadening, particularly near the phase transition.

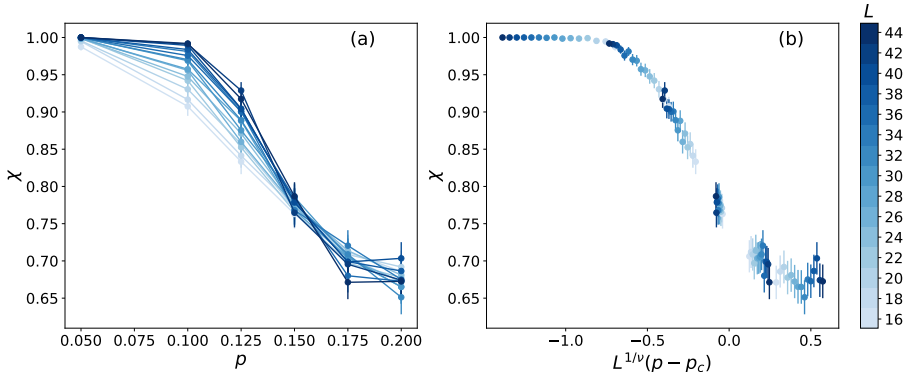

Figure 3.5: Raw and collapsed cross entropies for a 1D chain. (a) Cross entropy χ for a 1D chain with up to 22 physical qubits obtained from *ibm_sherbrooke*, corresponding to a system size of 44 qubits and with small systems ($L < 16$) removed. (b) Collapse of cross entropy curves near the critical point obtained by minimizing the scatter of all points to an unknown scaling function.

Comparing $\chi_{\rho=\sigma}$ (Fig. 3.2) with $\chi_{\rho\neq\sigma}$ (Fig. 3.3), we find that the former is often visibly smaller than the latter, particularly for the larger values of p we accessed in our experiments. On the other hand, as we show in Sec. 3.12 with rigorous arguments, one has the bound $\chi_{\rho=\sigma} \geq \chi_{\rho\neq\sigma}$ in Clifford circuits with a simple noise model, namely those that can be written as stabilizer operations and their probabilistic mixtures. These include the erasure errors we use in our classical numerical simulations (see Sec. 3.12). We attribute the violation of this bound to real device error, which necessarily involves, e.g., coherent and non-unital noise, that go beyond our simple noise model. Evidently, $\chi_{\rho=\sigma}$ is more sensitive to noise than when two different initial states are used.

3.7 Results for all-to-all connectivity

We finally present the experimental results for the all-to-all connectivity experiment. Theory predicts qualitatively similar results of the cross entropy to the 1D case, but the transition is in a different universality class [168]. The initial states used in this experiment are the same as in the 1D-chain case. Hardware executions of the raw, uncompressed circuits are difficult on current superconducting hardware, which typically only support gates acting on nearest-neighbor qubits. As a result, the raw circuits would require an excessive number of swap gates. With circuit compression, however, the average cost of executing the all-to-all circuits are the same as in the 1D-chain case, since the compressed circuits once again have $L/2$ qubits and at most $L/2$ mid-circuit measurements.

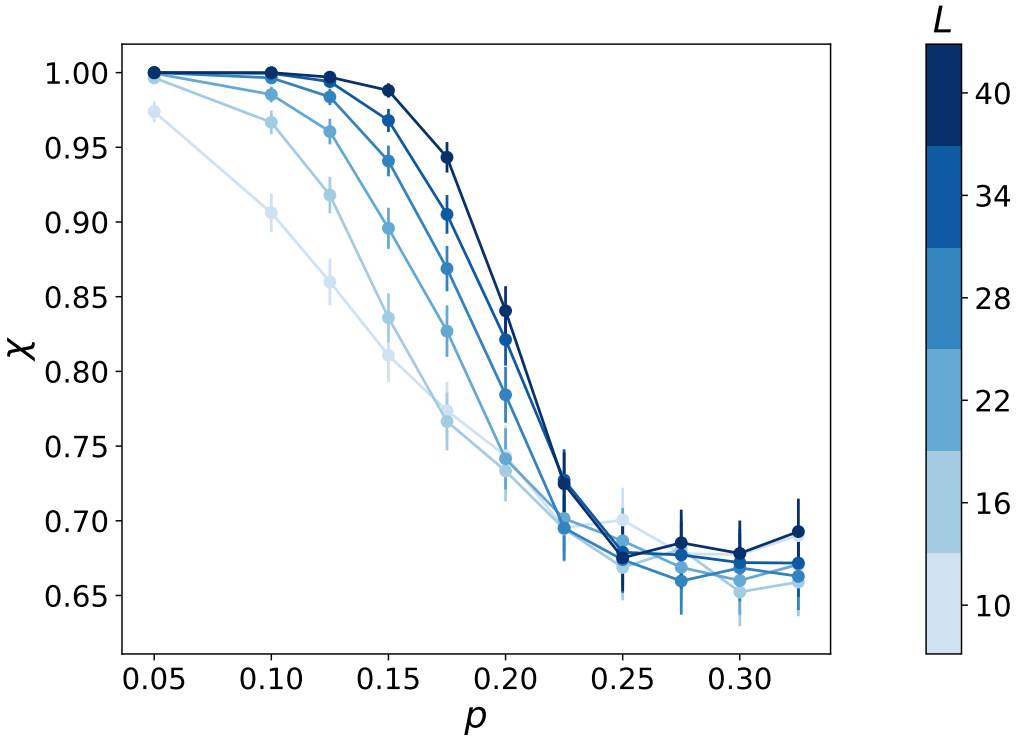

Figure 3.6: Cross entropy χ for infinite-dimensional systems with up to 20 physical qubits (corresponding to a system size of $L = 40$ qubits before compression) computed on *ibm_sherbrooke*.

Figures 3.6 and 3.7 shows that the phase transition is still observable despite the raw (uncompressed) all-to-all circuits requiring significantly more gates than in the 1D case. The qualitative features of χ in the all-to-all case are similar to the 1D-chain case, with larger values of χ for larger systems when $p < p_c$, crossing of all χ's at

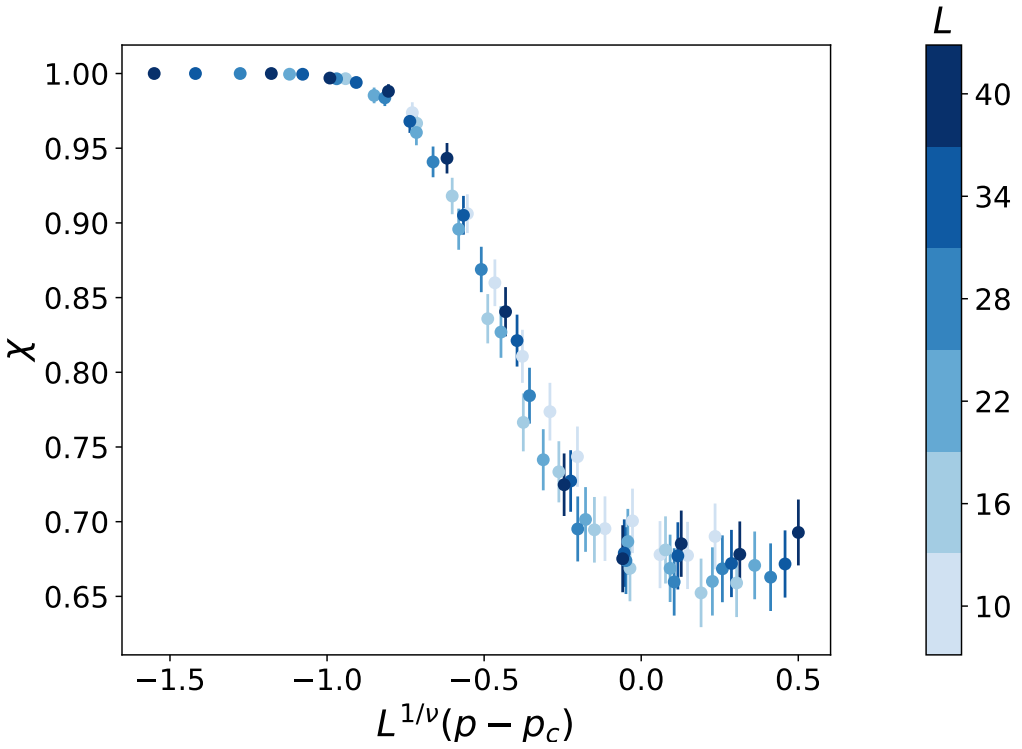

Figure 3.7: Collapse of cross entropy curves near the critical point obtained by minimizing the scatter of all points to an unknown scaling function. The fitting procedure gives a critical measurement rate of $p_c = 0.26 \pm 0.02$ and critical exponent $\nu = 1.9 \pm 0.4$.

the critical point, and a saturation to a constant for $p > p_c$. The critical values we extract from fitting to the finite size scaling form Eq. (3.15) are $p_c = 0.26 \pm 0.02$ and $\nu = 1.9 \pm 0.4$ at the 90% confidence level, with a different function F than in the 1D case. In particular, a mean-field analysis of all-to-all circuits [168] predicts $\nu \approx 2.5$, which lies within the range of confidence here . These results are consistent with our numerical simulations (see Sec. 3.12) for these system sizes,where a large uncertainty in these fitting parameters are also observed.

The increased value of p_c for the infinite-dimensional case compared to the 1D case is consistent with the intuitive picture that entanglement in a system with high connectivity is more stable to measurements that in one with low connectivity.

3.8 Fitting parameters ν and p_c by collapsing hardware data

Near the critical measurement rate p_c, the order parameter χ for different system sizes and under suitable rescaling is expected to collapse onto a single curve [124,

166, 167]. Quantitatively, this can be expressed as $\chi(L,p)$ collapsing to the same curve for all system sizes L when we suitably rescale both L and p:

$$\chi(L,p) = F\left[L^{1/\nu}(p-p_c)\right]. \tag{3.16}$$

The critical measurement rate depends on the microscopic details of the circuits, such as the encoding and bulk ratios, whereas the the critical exponent is independent of the microscopic circuit details and is the same for all systems in the same universality class [124, 166]. If the scaling function F was known, we could obtain the optimal p_c and ν, denoted by p_c^* and ν^*, by minimizing the residual sum of squares (RSS) over all data points:

$$p_c^*, \nu^* = \arg\min_{p_c,\nu} \sum_L \sum_p \left(F\left[L^{1/\nu}(p-p_c)\right] - \chi_{\text{exp}}(L,p)\right)^2, \tag{3.17}$$

where $\chi_{\text{exp}}(L,p)$ is the cross entropy obtained from the experiment for a system size L and measurement rate p. When the scaling function is unknown, we still find p_c^* and ν^* by minimizing an RSS, but instead use an interpolating function for our scaling function for a fixed L, followed by symmetrization over all L in order to prevent preferential treatment of any portion of the data [167, 60]. Our approach to fitting p_c and ν follows Ref. [60] with modifications due to there being only one critical exponent in our case, versus two critical exponents in Ref. [60]. We denote by $\mathcal{L}$ the set of system sizes used in the experiment and $\mathcal{P}_L$ the set of measurement rates used for a fixed L. For each $L \in \mathcal{L}$ and $p \in \mathcal{P}_L$, we first compute the rescaled controlled variable

$$q_L(p) = L^{1/\nu}(p-p_c). \tag{3.18}$$

We then construct an interpolating function for $\chi(L,p)$ from the rescaled experimental data, which we denote by $f_L(q)$. The interpolating function is used since the q values for different values of L are different, and the RSS is taken over points with identical q values. From numerical simulations, we expect the scaling function to decrease monotonically for increasing p [61]. To preserve this monotonicity, we use a piecewise cubic Hermite polynomial implemented in SciPy to construct the interpolating function [169]. We denote the set of q_L as $\mathcal{Q}_L$, $q_L^- = \min_{\mathcal{Q}_L}$ and $q_L^+ = \max_{\mathcal{Q}_L}$. Adapting the measure of goodness of fit from References [167] and [60], we define the loss function as

$$R(\nu,p_c) = \sum_{L\in\mathcal{L}} \sum_{\substack{L'\in\mathcal{L},\\ L'\neq L}} \sum_{\substack{q\in\mathcal{Q}_{L'},\\ q_L^-\le q\le q_L^+}} \left(f_L(q) - f_{L'}(q)\right)^2. \tag{3.19}$$

In the innermost summation, we constrain q by $q_L^- \leq q \leq q_L^+$ in order to avoid extrapolation of $f_L(q)$. Our reported best fit values of p_c and ν are then given by

$$p_c^*, \nu^* = \arg\min_{p_c,\nu} R(\nu, p_c). \tag{3.20}$$

Following References [60, 167], the errors for ν and p_c are given by the width of the minimum at level η:

$$\delta\nu_\pm = \eta\nu^* \left[2\log\frac{R(\nu^* \pm \eta\nu^*, p_c^*)}{R(\nu^*, p_c^*)}\right]^{-1/2} \tag{3.21}$$

$$\delta p_{c\pm} = \eta p_c^* \left[2\log\frac{R(\nu^*, p_c^* \pm \eta p_c^*)}{R(\nu^*, p_c^*)}\right]^{-1/2}. \tag{3.22}$$

Our final values of ν and p_c are then reported, setting η to the 10% level, as

$$\nu^* \pm \max(\delta\nu_+, \delta\nu_-) \tag{3.23}$$

$$p_c^* \pm \max(\delta p_{c+}, \delta p_{c-}). \tag{3.24}$$

In Figure 3.8(a) the cost function for the 1D chain is shown in a parameter space near the optimum. The reported uncertainties corresponding to the 90% confidence interval are the width of the minima when we increase the cost function by 10%. Figure 3.8(b) (3.8(c)) shows the cost function for the 1D chain when p_c (ν) is held fixed at its minimum and ν (p_c) is varied. Figure 3.9 shows the corresponding cost function for the all-to-all system.

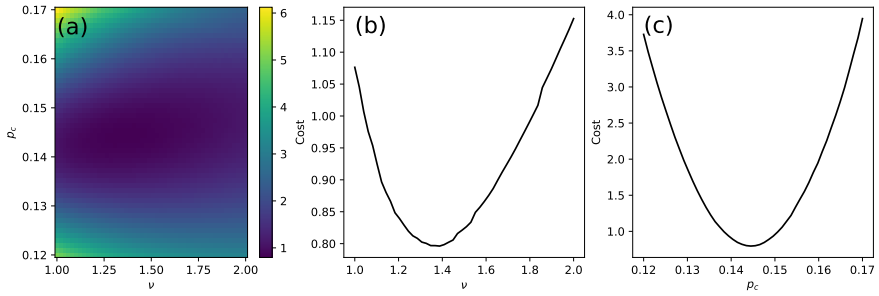

Figure 3.8: Cost function minimum in the region of the optimal solution for fitting critical values in the 1D chain experiment. (a) The cost function as defined in Equation (3.19) for the 1D chain with varying ν and p_c. (b) The cost function when p_c is held fixed at its optimal value and ν is varied. (c) The cost function when ν is held fixed at its optimal value and p_c is varied.

The low reported uncertainty in the value of p_c is a consequence of the cost function having a sharp minimum in the p_c direction (see Figures 3.8(b) and 3.9(b)), and

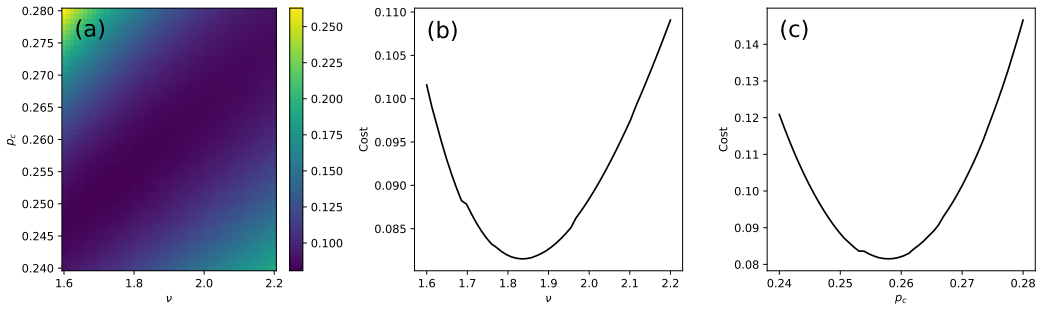

Figure 3.9: Cost function minimum in the region of the optimal solution for fitting critical values in the all-to-all connectivity experiment. (a) The cost function as defined in Equation (3.19) for the all-to-all system with varying ν and p_c. (b) The cost function when p_c is held fixed at its optimal value and ν is varied. (c) The cost function when ν is held fixed at its optimal value and p_c is varied.

the relatively large uncertainty for ν results from the cost function having a broad minimum (see Figures 3.8(c) and 3.9(c))

3.9 Calculation of error bars

In order for the linear cross entropy to be a scalable probe for measurement induced phase transitions, the number of circuits and and circuit evaluations required for a given (L, p) pair must be polynomial in L, p, and $1/\epsilon$, the error in estimating $\chi(L, p)$ from multiple samples. As shown in Reference [61], the number of samples can in fact be taken to be independent of L and p, and exhibits a linear dependence on N in $1/\epsilon$, where N is the number of circuits used. We can see this dependence explicitly in the calculation of the error bars reported in the main text, shown in the following.

For a given (L, p) pair, we use N randomly generated circuits and execute each circuit M times on IBM's quantum hardware, resulting in M different measurement outcomes. We calculate the cross entropy for each circuit i as

$$\chi_i = \frac{1}{M} \sum_{j=1}^{M} x_{ij}, \tag{3.25}$$

where x_{ij} is the j'th measurement bit string for the i'th circuit and is defined as

$$x_{ij} = \begin{cases} 1, & \text{if } x_{ij} \text{ can occur on } \sigma_i \\ 0, & \text{if } x_{ij} \text{ cannot occur on } \sigma_i \end{cases} . \tag{3.26}$$

Here, σ_i is the σ circuit corresponding to the i'th ρ circuit. We next calculate the standard error of the mean as

$$\epsilon_i = \frac{\hat{s}_i}{\sqrt{M}}, \quad \hat{s}_i^2 = \frac{1}{M-1}\sum_{j=1}^{M}(x_{ij} - \chi_i)^2. \tag{3.27}$$

We then compute the final estimate of the cross entropy as $\bar{\chi} = (1/N)\sum_{i=1}^{N}\chi_i$. The variance of $\bar{\chi}$ is given by

$$\epsilon^2 = \frac{1}{N}\sum_{i=1}^{N}\epsilon_i^2 \tag{3.28}$$

and the error bars reported in all figures are given by $\bar{\chi} \pm 1.96\epsilon$, representing the 95% confidence interval for the estimate of χ.

3.10 Error mitigation for hardware experiments

Dynamical decoupling

Dynamical decoupling (DD) is a quantum control technique employed in quantum computing to mitigate errors by taking advantage of time-dependent pulses [170, 171, 172, 173, 174, 175, 176]. In its simplest form, DD is implemented by sequences of X control pulses, whose effect is to protect qubits from decoherence due to low-frequency system-environment coupling. Here, we applied sequences of two X pulses (as in Ramsey echo experiments) to idle qubits. In Figure 3.10, we illustrate the impact of DD on the cross entropy, focusing on the $\rho = \sigma$ case. As seen, for $L \simeq 10$, DD increases the cross entropy towards the exact value of $\chi = 1$. However, the increase in χ is of order 0.01 whereas the difference between χ and 1 is of order 0.1 and, furthermore, it becomes less pronounced for $L \simeq 18$.

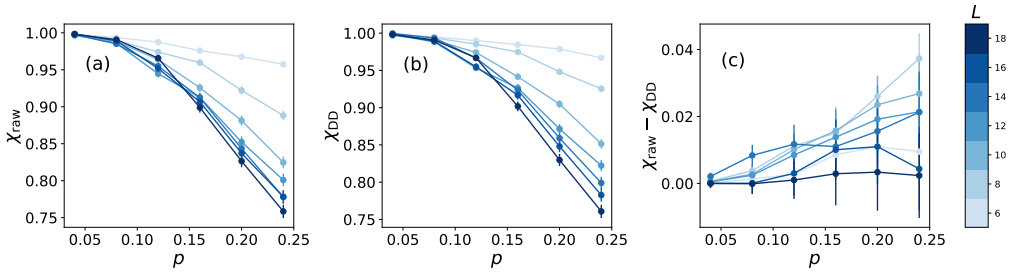

Figure 3.10: Cross entropy χ for chains of $L = 6$ to $L = 18$ qubits, with initial states $\rho = \sigma$, computed without (a) and with (b) dynamical decoupling, and difference between these two quantities (c).

Readout error mitigation

Readout error mitigation (ROEM) is a standard technique to compensate for errors incurred during qubit readout [177, 178]. We tested ROEM for small systems of up to $L = 14$ (7 physical qubits) and observed negligible differences between the readout error mitigated cross entropies and the unmitigated cross entropies, see Figure 3.11. Due to the negligible effects of ROEM, we did not use ROEM for any of the results presented in the main text.

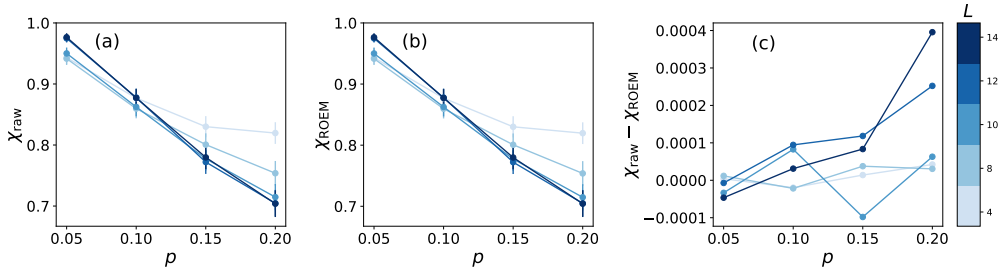

Figure 3.11: Effects of readout error mitigation on cross entropy for systems with up to 7 physical qubits. (a) The raw cross entropies without ROEM. (b) The cross entropies with ROEM applied. (c) The difference $\chi_{raw} - \chi_{ROEM}$, which shows that the differences between the raw and ROEM cross entropies are significantly smaller than the error bars for the raw cross entropies.

3.11 Resource analysis

Because this protocol makes use of measurement results without post-selection and does not require quantum state tomography, the quantum hardware resources required are significantly reduced compared to previous experiments. In particular, the total number of qubit-hours required was reduced from above 5200 qubit-hours in Ref. [60] to less than 8 qubit-hours in our demonstration while retaining the hardware implementation of mid-circuit measurements. Additionally, we were able to demonstrate this protocol using 22 hardware qubits, corresponding to uncompressed systems of up to 44 qubits. This size regime is presently inaccessible to protocols requiring post-selection.

With the use of circuit compression, we expect that larger systems with as many as 30 physical qubits could be accessed while still maintaining the fidelity of the current experiments. The limiting factor in our demonstration is the computational cost of the circuit compression, which scales polynomially with system size but is still too resource-intensive for circuits with $L > 50$ qubits. As this work focused on demonstrating the protocol on near-term hardware, we did notemphasize efficient

implementations of the classical circuit-compression algorithm; this task could be a focus of future work. To increase to larger system sizes, the bulk and encoding ratios can also be reduced from 3, used in our experiments, to as low as 1 while still maintaining a visible phase transition.

Improvement of the experimental performance of the processor, for instance by reducing cross-talk and introducing carefully tailored dynamical decoupling sequences may also allow us to explore the phase transition in even larger systems. Preliminary experiments including dynamical decoupling show some improvement in the fidelity obtained in the intermediate regime of 5 to 8 hardware qubits; however, for larger systems dynamical decoupling had little effect and so was not used in any of the experiments. We have also attempted a readout error mitigation for our $\rho \neq \sigma$ experiments in 1D, which did not change our results and was not applied to our data.

3.12 Simulated noisy data

In this section we provide classical numerical simulations as a reference for experimental data presented in the main text. All circuits considered here are drawn from the same ensemble as the experimental runs, and are simulated without compression.

1D circuits and a statistical mechanics picture

For the 1D case, we first choose $\rho = \sigma = (|0\rangle\langle 0|)^{\otimes L}$, as in Fig. 3.2. In the circuit, we insert an erasure channel at each spacetime location of the ρ-circuit with probability $q = 0.1\%$, while keeping the σ-circuit noiseless. The results are shown in Fig. 3.12(a), where we see a decrease in $\chi^{\text{noisy}}_{\rho=\sigma}$ when either L or p is increased. This trend is consistent with what we observe in Fig. 3.2. The data is consistent with the following functional form,

$$\chi^{\text{noisy}}_{\rho=\sigma} \propto \exp\left[-\alpha(p,q)\cdot L^2\right], \tag{3.29}$$

where $\alpha(p,q)$ is a nonzero coefficient depending on p and q, see Fig. 3.12(b).

Next we consider the $\rho \neq \sigma$ case, but instead with stabilizer initial states $\rho = \frac{1}{2^L}\mathbb{I}$ and $\sigma = (|0\rangle\langle 0|)^{\otimes L}$ to facilitate efficient classical simulation. In Fig. 3.13(a), we present numerical results obtained from a noiseless simulation. The overall trend of the results are comparable to those in Fig. 3.3. The data collapse in Fig. 3.13(b) is performed with $p_c = 0.16$ and $\nu = 1.33$, as consistent with Ref. [61].

We also perform a noisy simulation for $\rho = \frac{1}{2^L}\mathbb{I}$ and $\sigma = (|0\rangle\langle 0|)^{\otimes L}$, where we insert an erasure channel at each spacetime location of the ρ-circuit with probability

$q = 0.1\%$. The numerical results are shown in Fig. 3.14. As we anticipate from statistical mechanics arguments (see Ref. [61] and below), for any finite noise rate, the cross entropy will be suppressed to zero for all value of p, in the thermodynamic limit. For small system sizes (before the cross entropy is reduced to zero) the curves will instead appear to cross at a smaller value of p_c. Indeed, the best fit for p_c has now shifted to a smaller value, $p_c \approx 0.14$ (whereas we use the same value for ν), close to the one used for fitting in the main text.

Statistical mechanics picture

The qualitative behavior the results in Fig. 3.14 can be understood from a mapping to statistical mechanics models, which we briefly describe here. (We refer the reader to Ref. [61] and references therein for further details.) Recall that

$$\chi := \mathbb{E}_C \chi_C = \mathbb{E}_C \frac{\sum_{\mathbf{m}} p^{\rho}_{\mathbf{m}} p^{\sigma}_{\mathbf{m}}}{\sum_{\mathbf{m}} (p^{\sigma}_{\mathbf{m}})^2} = \mathbb{E}_C \frac{\sum_{\mathbf{m}} \mathrm{Tr}\left(C_{\mathbf{m}}(\rho)\right) \cdot \mathrm{Tr}\left(C_{\mathbf{m}}(\sigma)\right)}{\sum_{\mathbf{m}} (\mathrm{Tr}\left(C_{\mathbf{m}}(\sigma)\right))^2}$$
$$= \mathbb{E}_C \frac{\sum_{\mathbf{m}} \mathrm{Tr}\left(C^{\otimes 2}_{\mathbf{m}}(\rho \otimes \sigma)\right)}{\sum_{\mathbf{m}} \mathrm{Tr}\left(C^{\otimes 2}_{\mathbf{m}}(\sigma \otimes \sigma)\right)}. \tag{3.30}$$

Here $C_{\mathbf{m}}(\rho)$ denotes the resultant state when unitaries and projective measurements (labeled by the measurement record $\mathbf{m}$) from C are applied to the initial state ρ. It is easier to study the following proxy quantity, which is an approximation of χ by

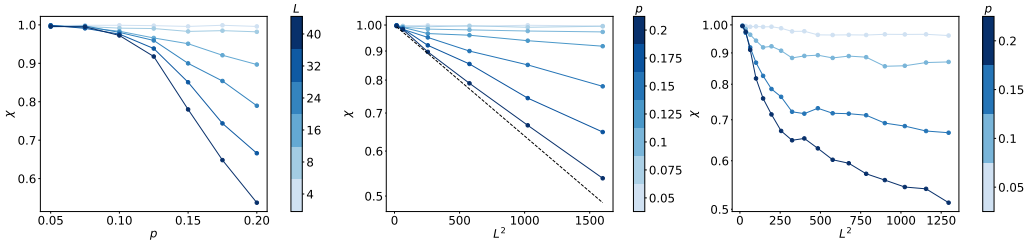

Figure 3.12: Noisy numerical simulations for the 1D chain using erasure noise. (a) Results from noisy numerical simulations of Clifford circuits in 1D, for system sizes $L \leq 40$. We take the initial states $\rho = \sigma = (|0\rangle\langle 0|)^{\otimes L}$ as in Fig. 3.2, and randomly insert an erasure channel at each spacetime location of the ρ-circuit with probability $q = 0.1\%$. (b) We find the data consistent with the functional form in Eq.(3.29). (c) Experimentally obtained χ. The non-linear behaviour may be caused due to coherent errors or other noise sources not captured by an erasure channel.

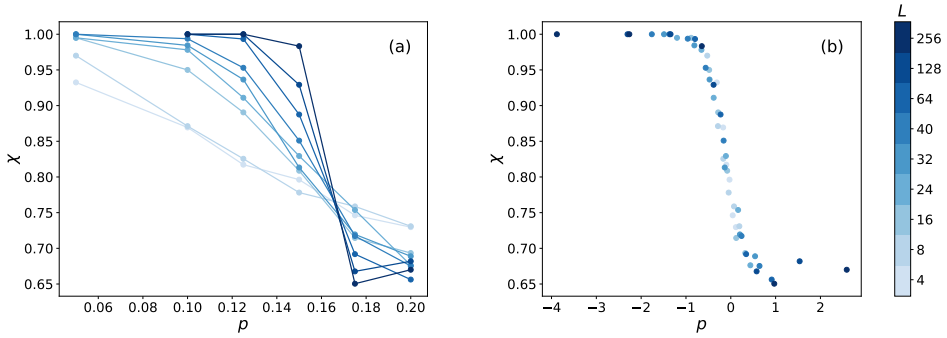

Figure 3.13: Noisy numerical simulations for the 1D chain using erasure noise. (a) Results from noiseless numerical simulations of Clifford circuits in 1D, for system sizes $L \leq 256$. In our simulation, we take $\rho = \frac{1}{2^L}\mathbb{I}$ and $\sigma = (|0\rangle\langle 0|)^{\otimes L}$, as in Ref. [61]. (b) When fitting the data to the scaling form in Eq. (3.15), we obtain $p_c \approx 0.16$ and $\nu \approx 1.33$, as consistent with Ref. [61].

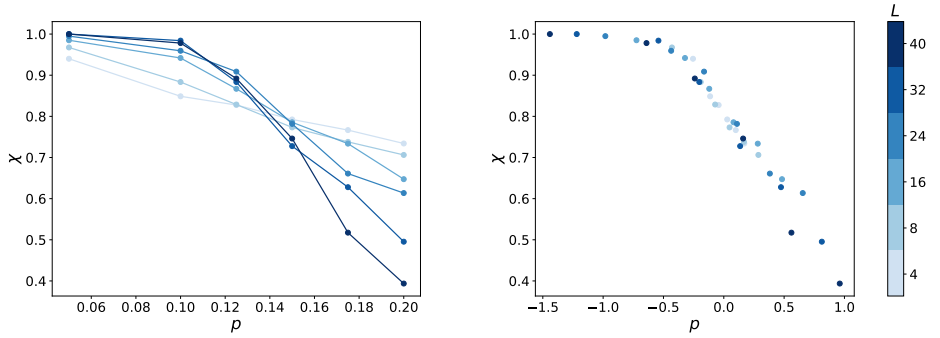

Figure 3.14: Noisy numerical simulations for the 1D chain using erasure noise. (a) Results from noisy numerical simulations of Clifford circuits in 1D, for system sizes $L \leq 40$. We take the same initial states ρ and σ as in Fig. 3.13, and randomly insert an erasure channel at each spacetime location of the ρ-circuit with probability $q = 0.1\%$. (b) When fitting the data to the scaling form in Eq. (3.15), we use $p_c \approx 0.14$ and $\nu \approx 1.33$ as obtained from Fig. 3.4, where we find consistency.

averaging the numerator and the denominator separately over C,

$$\overline{\chi} = \frac{\mathbb{E}_C \sum_{\mathbf{m}} \mathrm{Tr}\left(C_{\mathbf{m}}^{\otimes 2}(\rho \otimes \sigma)\right)}{\mathbb{E}_C \sum_{\mathbf{m}} \mathrm{Tr}\left(C_{\mathbf{m}}^{\otimes 2}(\sigma \otimes \sigma)\right)}. \tag{3.31}$$

For C a brickwork circuit with local 2-qubit random unitary gates forming a 2-design, the averages can be performed. As a result, the numerator and the denominator will both take the form of a partition function of the Ising model on a triangular lattice, where the Boltzmann weights can be explicitly written down [179, 148, 129, 180, 181]). The two partition functions are identical in the bulk, and only differ in

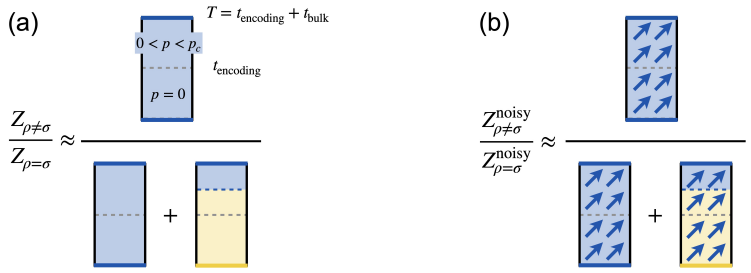

Figure 3.15: Mapping $\overline{\chi}$ defined in Eq. (3.32) to quantities in an effective Ising model, when the circuit is (a) noiseless and (b) noisy. See the text for more details. In both figures the blue color represents spins pointing in the "+" direction, the yellow color represents spins pointing in the "−" direction, and the black color represents a "free" boundary condition, where the spins can point in either direction.

their boundary conditions (coming from the difference in initial states). Following Ref. [61], we denote them $Z_{\rho\neq\sigma}$ and $Z_{\rho=\sigma}$, respectively.

In all our circuits we choose ρ and σ to be tensor products of onsite density matrices, and let them be different states. We also take the circuit to have a purely-unitary "encoding" stage without measurements, before measurements take place (see Fig. 3.1). Within these circuits, $\overline{\chi} = Z_{\rho\neq\sigma}/Z_{\rho=\sigma}$ corresponds to the partition function ratio shown in Fig. 3.15(a). Each term lives in a rectangular geometry, with the lower half an Ising model at zero temperature (corresponding to the encoding stage), and the upper half at finite temperature [61]. The blue color denotes a "+" boundary condition, and the yellow color denotes a "−" one. The numerator $Z_{\rho\neq\sigma}$ has a boundary condition where both the top and bottom spins are fixed to be +, whereas $Z_{\rho=\sigma}$ has an additional contribution where the bottom boundary condition is also "−". Thus,

$$\overline{\chi}_{\rho\neq\sigma} = \frac{Z_{\rho\neq\sigma}}{Z_{\rho=\sigma}} = \frac{1}{1 + Z_{+-}/Z_{++}}. \tag{3.32}$$

The $p < p_c$ phase of circuit maps to the the ferromagnetic phase of the Ising magnet, where $-\ln(Z_{+-}/Z_{++})$ is the free energy of a horizontal domain wall separating the bottom and the top (see Fig. 3.15(a)), which diverges with L, therefore $\overline{\chi} \to 1$. On the other hand, in the $p > p_c$ "paramagnetic" phase the domain wall free energy vanishes, so $Z_{+-}/Z_{++} \to 1$ and $\overline{\chi} \to 1/2$. We see that the numerical value of $\overline{\chi}$ in the $p > p_c$ phase differs from our numerical results, due to the annealed average.

The Ising picture is also useful for a qualitative understanding of the behavior of linear cross entropy in the presense of noise. For simplicity, we take the the noise

to be a random erasure at each spacetime location. The cross entropy now reads

$$\chi := \mathbb{E}_{C,\mathcal{N}} \chi_{C,\mathcal{N}} = \mathbb{E}_{C,\mathcal{N}} \frac{\sum_{\mathbf{m}} \mathrm{Tr}\left((C'_{\mathbf{m}} \otimes C_{\mathbf{m}})(\rho \otimes \sigma)\right)}{\sum_{\mathbf{m}} \mathrm{Tr}\left(C_{\mathbf{m}}^{\otimes 2}(\sigma \otimes \sigma)\right)}. \tag{3.33}$$

Here the circuit C' is obtained from C by inserting erasure noise (denoted $\mathcal{N}$) at random spacetime locations, which in general turns pure states into mixed states. Similarly, we define

$$\overline{\chi}^{\text{noisy}}_{\rho\neq\sigma} = \frac{\mathbb{E}_{C,\mathcal{N}} \sum_{\mathbf{m}} \mathrm{Tr}\left((C'_{\mathbf{m}} \otimes C_{\mathbf{m}})(\rho \otimes \sigma)\right)}{\mathbb{E}_{C} \sum_{\mathbf{m}} \mathrm{Tr}\left(C_{\mathbf{m}}^{\otimes 2}(\sigma \otimes \sigma)\right)} = \frac{Z^{\text{noisy}}_{\rho\neq\sigma}}{Z_{\rho=\sigma}}. \tag{3.34}$$

This quantity is similar to our of experimental data in Fig. 3.3. We can also consider the following ratio

$$\overline{\chi}^{\text{noisy}}_{\rho=\sigma} = \frac{Z^{\text{noisy}}_{\rho=\sigma}}{Z_{\rho=\sigma}}, \tag{3.35}$$

which approaches 1 as the noise rate vanishes, and is similar to Fig. 3.2. Both $Z^{\text{noisy}}_{\rho\neq\sigma}$ and $Z^{\text{noisy}}_{\rho=\sigma}$ can be obtained from their noiseless versions by applying a magnetic field in the "+" direction everywhere in the system, breaking the Ising symmetry and also destroy the phase transition.

Nevertheless, we may still make a prediction from the stat mech picture for noisy data in a finite-size system. As we illustrate in Fig. 3.15(b), the following ratio should always be upper bounded by 1,

$$\frac{\overline{\chi}^{\text{noisy}}_{\rho\neq\sigma}}{\overline{\chi}^{\text{noisy}}_{\rho=\sigma}} = \frac{Z^{\text{noisy}}_{\rho\neq\sigma}}{Z^{\text{noisy}}_{\rho=\sigma}} = \frac{1}{1 + Z_{+-}(h>0)/Z_{++}(h>0)} \leq 1, \tag{3.36}$$

as the Ising partition functions remain positive under the erasure channel.

All-to-all circuit

We perform classical numerical simulations for circuits with all-to-all connectivity, taking the same initial states as our 1D simulations. The results are shown in Fig. 3.16, 3.17. We fit both noiseless and noisy data to the scaling form in Eq. (3.15). From the noiseless simulation of $L \leq 256$ we obtain fits $p_c \approx 0.33$ and $\nu \approx 2.50$. In particular, the critical exponent $\nu \approx 2.50$ agrees with a mean-field analysis as well as numerical simulations from Ref. [168]. We also observe that if we only include data from $L \leq 40$, then both the parameters here $(p_c, \nu) \approx (0.33, 2.50)$, and the

best fits obtained from experimental data $(p_c, \nu) \approx (0.26, 1.90)$ (see Fig. 3.6), will result in high quality data collapses (data not shown). This is consistent with our observation of a large uncertainty in the fitting parameters in Fig. 3.6.

On the other hand, from our noisy data at noise rate $q = 0.1\%$, we obtain $p_c \approx 0.20$ and $\nu \approx 0.80$. Recall that the same noise model and noise rate produced Fig. 3.14, which are comparable to experimental results in 1D. This suggests that noise affects the data strongly in all-to-all connectivity, and our experimental data cannot be fully captured by the simple simulated noise model.

3.13 Discussions

Our results show that MIPTs can be studied efficiently for systems with different connectivities on near-term superconducting quantum hardware, when restricted to Clifford circuits with an arbitrary initial state. The cross-entropy protocol used in this chapter eliminates both of the exponential bottlenecks in previous studies of MIPTs on superconducting hardware [60] while preserving the mid-circuit measurements in the bulk of the circuit, providing a benchmark for the quality of mid-circuit measurements in near-term quantum hardware. In future work, this protocol may be extended to extract other critical exponents using a different circuit structure [61], or to detect other related phenomena [182, 183, 144, 184, 185].

A technique we used throughout this work is Clifford circuit compression, which takes a circuit with non-stabilizer initial states and outputs a gate-efficient representation for it. Circuit compression allows us to study systems larger than the number

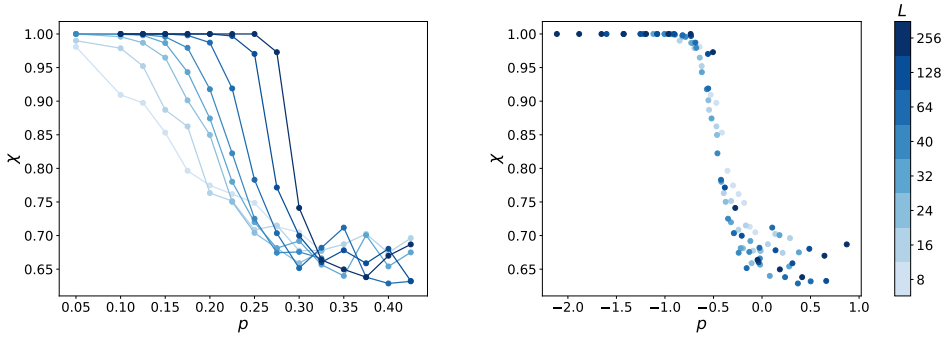

Figure 3.16: Noisy numerical simulations for the all-to-all connectivity system using erasure noise. (a) Results from noiseless numerical simulations of Clifford circuits with all-to-all connectivity, for system sizes $L \leq 256$. In our simulation, we take $\rho = \frac{1}{2^L}\mathbb{I}$ and $\sigma = (|0\rangle\langle 0|)^{\otimes L}$, identical to our choices in Fig. 3.13. (b) When fitting the data to the scaling form in Eq. (3.15), we obtain $p_c \approx 0.33$ and $\nu \approx 2.50$.

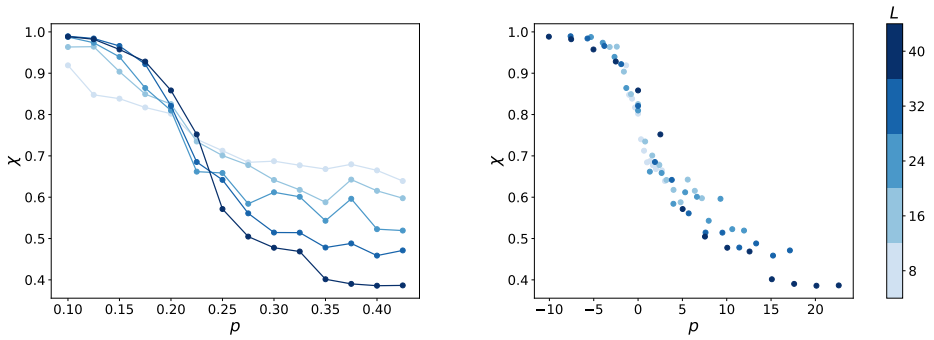

Figure 3.17: Noisy numerical simulations for the all-to-all connectivity system using erasure noise. (a) Results from noisy numerical simulations of Clifford circuits with all-to-all connectivity, for system sizes $L \leq 40$. We take the same initial states ρ and σ as in Fig. 3.16, and randomly insert an erasure channel at each spacetime location of the ρ-circuit with probability 0.1%. (b) When fitting the data to the scaling form in Eq. (3.15), we find $p_c \approx 0.20$ and $\nu \approx 0.80$.

of available hardware qubits, while minimizing the number of measurements, which is the slowest process of our circuits. Although we do not have a way to analyze or control the effect of noise in the compressed circuit, from the cross entropy data for $\rho \neq \sigma$ we can extract critical exponents that are evidently comparable to classical simulations and theoretical predictions. On the other hand, the same circuit but with $\rho = \sigma$ appear to be more sensitive to noise, which cannot be explained by a simple incoherent noise model. It will be an interesting future direction to explore the effects and the description of real device noise on the critical properties, and conversely, the extent to which a phase transition in cross entropy can be informative of experimental conditions and changes, such as dynamical decoupling, for large systems where process tomography is too costly.

Chapter 4

QUANTUM ADVANTAGE IN ADVERSARIALLY ROBUST MACHINE LEARNING

The previous chapters have demonstrated the applicability of noisy quantum computers for the physical problems of open quantum system simulation and studying measurement induced phase transitions. In the near term, noisy quantum computations may be useful for these and similar areas. The ultimate long term goal of quantum computing, however, is to scale quantum computers and reduce errors to a level which allows for large scale fault tolerant quantum computing. Among the potential applications of fault tolerant quantum computing is to speed up classical machine learning algorithms.

The growth of machine learning research in the past several years has led to the widespread real world deployment of machine learning algorithms. As they continue to be used throughout society, the safety of these algorithms has become an important topic. As a result, the study of adversarial attacks and defenses has become increasingly important. Many studies have been done on whether quantum learning algorithms are also susceptible to adversarial attacks, and whether they can offer additional defense over classical algorithms; however, to date these studies have mostly been heuristic in nature. In this chapter we provide the first rigorous quantum advantage for defending against adversarial attacks. In particular, we construct a binary classification learning problem for which any efficient classical algorithm cannot successfully defend against a specific adversarial attack, but for which a quantum algorithm can defend against and efficiently classify data.

4.1 Introduction

The past decade has seen large growth in machine learning research owing in large part to the near-human level of performance of neural network architectures for tasks ranging from image recognition and classification [186, 187, 188] to natural language processing [189, 190, 191] to generative modeling [192, 193]. In parallel, quantum computing has seen a similar growth due to its potential for applications such as quantum simulation of physics and chemistry [194, 195] and cryptography [196, 10]. Given the achievements and prospects of both quantum computing and machine learning, a natural question to ask is what, if any, advantage is there

in using quantum computers for classical machine learning tasks. So far, it has been shown that for a variety of learning frameworks, one can use cryptographic assumptions to construct learning problems for which quantum learners *provably* outperform classical learners [197, 198]. However, whether there exists a rigorous quantum advantage for practically relevant learning problems is unknown. One learning problem which has not been studied from the perspective of cryptographic assumptions is that of adversarially robust learning. In this framework, one is required to learn a model which performs accurately even in the presence of an adversary which can perturb the inputs in any way. A typical motivating example is when an adversary can perturb an image in a way which is imperceptible to the human eye, but causes a highly accurate classifier to mislabel the image.

Szegedy et al. [199] were the first to show that well performing neural networks for classification are susceptible to adversarial attacks. Attacks on automatic speech recognition systems and control systems [200, 201] and autonomous vehicle vision systems [202] were subsequently discovered. An analysis of adversarial attacks in linear models was given by Goodfellow et al. [203] and provides insight into why these attacks work in neural networks. By perturbing each pixel in an image proportionally to the sign of the gradient of the loss function, a small perturbation can result in a model misclassifying the perturbed input despite the perturbation being imperceptible to the human eye [203]. Since the seminal papers of Szegedy et al. [199] and Goodfellow et al. [203], various heuristic defenses have been constructed to protect neural networks from adversarial attacks. These defenses include the use of perturbed data into the training set [202] and a framework known as distillation of knowledge [204, 205] among others [206]. Despite the abundance of heuristic methods to defend against adversarial attacks, a rigorous understanding of which defenses work against which attacks and how well they work is lacking in the literature.

Recent results [207, 208] suggest there are computational barriers to learning classifiers that are robust to adversarial perturbations. In Reference [207], a learning problem is constructed such that non-robust classification is easy but robust classification is hard as a result of the computational indistinguishability of the output of pseudorandom generators from the uniform distribution. Although Reference [207] provides evidence for the hardness of robust learning as a result of computational limitations, the problem constructed there is not practically relevant; for real world problems, the hardness of robust learning may also be rooted in computational

limitations but so far this connection has not been made in the literature.

Adversarial attacks and defenses on quantum learning algorithms have been well studied in the literature. Quantum principal component analysis and ensemble methods have have been used to harden quantum classifiers against poisoning attacks [209]. Adding noise to quantum classifiers has been shown to increase robustness to attacks [210, 211] with robustness bounds which improve as the level of noise increases [211]. Liu and Wittek showed that the classification of quantum states using a quantum classifier is vulnerable to adversarial attacks as a result of the concentration of measure phenomenon in high dimensional Hilbert spaces [212], although encodings of realistic classical data has been shown, also using concentration of measure arguments, to exhibit increased robustness to attacks [213]. Alternative ways to obtain robustness bounds include linking quantum classification to quantum hypothesis testing [214], which also provides a protocol for assessing the robustness of classifiers to both random and adversarial noise and a protocol to determine whether a classifier outputs the same class for a perturbed state as an unperturbed state without requiring access to the unperturbed state. By providing access to states similar to those used in quantum PAC learning [215], classifiers can be constructed which are provably adversarially robust [216]. Numerical experiments have shown that quantum neural networks (QNNs) are susceptible to the same attacks as classical neural networks and that the same defenses can work in both the quantum and classical setting [212], although QNNs can exhibit enhanced robustness over classical adversaries by learning features that their classical counterparts do not [217]. Similar experiments have also been carried out on quantum hardware with up to 10 qubits, confirming the numerical results [218].

In this paper, we show constructively, under standard cryptographic assumptions, that there exists a class of classification problems for which any efficient classical learner is susceptible to an adversarial attack but for which a quantum learner can successfully defend against the same attack. Our work builds directly on top of the works of Bubeck et al. [207] and Degwekar et al. [208], and our contribution is to instantiate their learning problem in a way which allows for easy robust classification when given access to a quantum computer. We emphasize here that although our results provide an adversarial learning problem which exhibits a quantum advantage, the learning problem is not practically relevant. The complexity separation we provide is based on cryptographic assumptions and not real world learning problems. Nonetheless, our construction provides a necessary condition

for quantum advantages in robust learning and we believe it is an interesting proof of principle for adversarially robust quantum machine learning.

In Section 4.2, based on Ref. [207], we introduce the concept of adversarially robust machine learning and how hard robust learning tasks can be constructed from cryptographic primitives. In Section 4.4, based on Ref. [208], we illustrate how classical error correcting codes can be used on top of cryptographic primitives to allow for increased robustness. In Section 4.6, containing our main results, we introduce a learning task which, under widely accepted cryptographic assumptions, is infeasible for any efficient classical learner to learn *robustly*, but for which a quantum learner can efficiently learn an adversarially robust hypothesis. Finally, in Section 4.11 we discuss our results.

4.2 Adversarially robust classification from pseudo-random generators

Robust binary classification

We first define binary classification for the non-robust case for a pair of distributions $(\mathcal{D}_0, \mathcal{D}_1)$ supported on $\mathbb{Z}_2^n$ [207].

Definition 4.2.1 (Binary classification problem)**.** *A binary classification problem over $\mathbb{Z}_2^n$ is defined via a tuple $(\mathcal{D}_0, \mathcal{D}_1)$, where both $\mathcal{D}_0$ and $\mathcal{D}_1$ are distributions supported on $\mathbb{Z}_2^n$. For any $\epsilon \in (0, 1/2)$, a function $f : \mathbb{Z}_2^n \to \{0, 1\}$ is an ϵ-accurate solution to $(\mathcal{D}_0, \mathcal{D}_1)$ if*

$$\mathbb{P}_{X\sim\mathcal{D}_0}[f(X) = 0] \geq 1 - \epsilon \quad \text{and} \quad \mathbb{P}_{X\sim\mathcal{D}_1}[f(X) = 1] \geq 1 - \epsilon. \tag{4.1}$$

We note that a binary classification problem as defined above may not have a solution f which satisfies Equation (4.1). A trivial example is if $\mathcal{D}_0 = \mathcal{D}_1$.

We define a learning algorithm (for the binary classification problem) as

Definition 4.2.2 (PAC binary classification algorithm)**.** *We say that a learning algorithm $\mathcal{A}$ is an (ϵ, δ) probably approximately correct (PAC) learner for a family of binary classification problems $\mathcal{D} = \{(\mathcal{D}_0^i, \mathcal{D}_1^i)\}$ if, for all i, for any $\delta \in (0, 1)$ and $\epsilon \in (0, 1/2)$, when given sample access to $\mathcal{D}_0^i$ and $\mathcal{D}_1^i$, algorithm $\mathcal{A}$ outputs, with probability at least $1 - \delta$, an ϵ-accurate solution to $(\mathcal{D}_0^i, \mathcal{D}_1^i)$.*

When learning a binary classifier, and for many learning problems in general, there are several distinct notions of efficiency that are important when comparing learning algorithms. The query complexity of an (ϵ, δ) PAC learner the number of samples

required to output a classifier. The time complexity of learning is the runtime, as a function of $n, 1/\delta$, and $1/\epsilon$, of the learning algorithm to output a hypothesis. The time complexity of the hypothesis is the runtime, as a function of $n, 1/\delta$ and $1/\epsilon$, required to output a label. All three of the aforementioned complexities are important when evaluating the complexity of a learning algorithm. We say that a learning algorithm is efficient only if its query and time complexity, as well as the time complexity of the hypothesis, is $O(\text{poly}(n, 1/\epsilon, 1/\delta))$.

The previous definitions required only that the classifier is able to classify individual points accurately. In the presence of an adversary, however, it is not enough to classify individual points accurately but also all points reachable by an adversary. In this paper we consider an adversary that can perturb points arbitrarily within an λ-ball surrounding a data point. We therefore define an λ-robust version of the above formulation which requires that not only points from each distribution are correctly classified with high probability, but also that all points within an λ-ball are correctly classified:

Definition 4.2.3 (λ-robust binary classification problem)**.** *A λ-robust binary classification problem is defined via a tuple $(\mathcal{D}_0, \mathcal{D}_1)_\lambda$ where both $\mathcal{D}_0$ and $\mathcal{D}_1$ are distributions supported on $\mathbb{Z}_2^n$. For any $\epsilon \in (0, 1/2)$, a function $f : \mathbb{Z}_2^n \to \{0, 1\}$ is an ϵ-accurate solution to $(\mathcal{D}_0, \mathcal{D}_1)_\lambda$ if*

$$\mathbb{P}_{X\sim\mathcal{D}_0}[f(Y) = 0 \;\forall\, Y \in B(X, \lambda)] \geq 1 - \delta \quad \text{and} \tag{4.2}$$

$$\mathbb{P}_{X\sim\mathcal{D}_1}[f(Y) = 1 \;\forall\, Y \in B(X, \lambda)] \geq 1 - \delta, \tag{4.3}$$

where $B(x, \lambda) = \{z \in \mathbb{Z}_2^n \;:\; d(x, z) \leq \lambda\}$ and $d(\cdot, \cdot)$ is the Hamming metric on $\mathbb{Z}_2^n$.

We note that in Reference [207], the distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ are supported over $\mathbb{R}^n$ allowing for perturbations which are small with respect to the L_2 norm on $\mathbb{R}^n$. In our definition, we restrict our distributions to be supported on $\mathbb{Z}_2^n$ and only allow for bit flip perturbations. We impose these restrictions in order to simplify our results although generalisation to $\mathbb{R}^n$ is possible. Additionally, we note that Definition 4.2.3 reduces to the non-robust case when we set $\lambda = 0$.

We finally define a λ-robust PAC learning algorithm as

Definition 4.2.4 (λ-robust PAC binary classification algorithm)**.** *We say that a learning algorithm $\mathcal{A}$ is λ-robust, (ϵ, δ) PAC learner for a family of binary classification problem $\mathcal{D} = \{(\mathcal{D}_0^i, \mathcal{D}_1^i)_{\lambda_i}\}$ if, for all i, for all $\delta \in (0, 1)$ and $\epsilon \in (0, 1/2)$, when*

given sample access to $\mathcal{D}_0^i$ and $\mathcal{D}_1^i$, algorithm $\mathcal{A}$ outputs, with probability of at least $1-\delta$, an ϵ-accurate solution to $(\mathcal{D}_0^i, \mathcal{D}_1^i)_{\lambda_i}$.

The sample and computational complexities are defined as for non-robust classification, except the complexities are additionally functions of λ, which we consider a function of n. λ-robust binary classification requires that not only points from each distribution are correctly classified with high probability, but also that *every* point in an λ-ball around a given point from either distribution is correctly classified with high probability. When considering an adversary rather than random noise, we require that every point in an λ-ball is correctly classified so that even in the worst case, where the adversary can perturb the data in any way provided the size of the perturbation is less than λ, we are still able to accurately classify points. In the first inequality in Equation 4.2.3, this is represented by the function f being identically 0 on the λ-ball so that no perturbation smaller than λ can cause a misclassification. In the second inequality, f must be identically 1 on the λ-ball so that no perturbation smaller than λ can cause a misclassification.

Hard robust-classification problems

As described in Ref. [207] and Section 4.9, we can construct a learning problem for pseudo-random generators (PRGs) and uniform distributions such that a function f exists such that non-robust classification is efficient whereas λ-robust classification is possible but outputting a $\Theta(\sqrt{n})$-robust classifier is computationally inefficient.

A PRG is family $\{G_n\}$ of deterministic functions

$$G_n \ : \ \{0,1\}^n \to \{0,1\}^m, \quad m = \text{poly}(n) \tag{4.4}$$

such that the output distribution of G_n, on uniformly random inputs $\{0,1\}^n$, is computationally indistinguishable from the uniform distribution on $\{0,1\}^m$. See Section 4.7 for rigorous definitions.

Since $\mathcal{D}_0$ is the output of a PRG and $\mathcal{D}_1$ the uniform distribution, efficient classification of points as in Eq. (4.2) implies efficient distinguishability between the output of the PRG and the uniform distribution, contrary to the definition of a PRG. By prepending the data labels, the classification can be made efficient in the non-robust case. The adversary then randomly flips the first bit of the input, rendering the robust classification as difficult as inverting the PRG [207]. We formalize the classification task as follows.

Construction 4.2.1 (Hard robust-classification problem from PRGs)**.** *Let* $G : \{0,1\}^n \to \{0,1\}^m$ *be a pseudorandom generator. We construct a hard robust-classification problem (in the sense of Definition 4.2.3) from* G *by defining the two distributions* $\mathcal{D}_0$ *and* $\mathcal{D}_1$*.* $\mathcal{D}_0$ *is defined as the distribution of* $(0, G(X))$ *where* X *is distributed uniformly over* $\{0,1\}^n$*.* $\mathcal{D}_1$ *is the distribution of* $(0, Y)$*, where* Y *is distributed uniformly over* $\{0,1\}^m$*.*

In the above construction, if the first bit is flipped, computational indistinguishability of $G(X)$ and Y implies that efficient classification is not possible. In the following sections, we will instantiate the PRG G using specific cryptographic assumptions.

The complexity separation between robust and non-robust classifiers as constructed above relies crucially on the computational indistinguishability between the distribution of the output of the PRG and the uniform distribution. PRGs which rely on cryptographic assumptions, such as the Decisional Diffie-Hellman (DDH) assumption over the quadratic residues modulo a safe prime, are thought to provide the necessary randomness required in the above constructions as the DDH assumption is widely believed to hold in this case [219]. However, although no efficient classical algorithm is known which can distinguish between the outputs of the DDH PRG and the uniform distribution, a simple application of Shor's algorithm for discrete logarithms [10] would be sufficient to distinguish between the pseudo-random and uniformly random bits. Our construction of an adversarially robust machine learning problem which exhibits a rigorous quantum-classical separation is based on the application of Shor's algorithm to break the PRGs used in the construction of hard robust learning problems.

4.3 A quantum-classical separation with constant robustness

As a warm-up, we describe in this section how to obtain a quantum advantage for a robust learning problem when the adversary is allowed to flip $O(1)$ bits. The robust learning problem follows directly from Construction 4.2.1 described in [207]. We use the fact that there exists, under certain cryptographic assumptions, families of functions $G = \{G_n\}$ which are provably pseudorandom but which are easy to invert if given access to a fault-tolerant quantum computer. More details are provided in Section 4.6 and Section 4.8, but for the purpose of this section we note that if there exists an efficient algorithm for computing discrete logarithms then it is efficient to determine whether a bit string is in the image of the PRG G or not.

We now describe an efficient algorithm to classify bit strings in the presence of an adversary which can flip $O(1)$ bits. In particular, we choose an adversary which always sets the first bit to 0 and which leaves the rest of the bit string unchanged. The input to the algorithm is thus a bit a string of the form $0||x$. We assume that we have access to an efficient quantum algorithm $\mathcal{A}$ which on input x outputs 0 if $x \in \text{Image}(G)$ and 1 if $x \notin \text{Image}(G)$. The existence of such an algorithm $\mathcal{A}$ is shown in Section 4.6. On input $0||x$, the classification algorithm simply outputs $\mathcal{A}(x)$. If $\mathcal{A}$ is efficient and outputs the correct label with high probability, then the classification is efficient and accurate.

4.4 Increasing robustness via error correcting codes

Even if there exists an algorithm which can efficiently distinguish between outputs from the PRG and the uniform distribution, the above construction can only efficiently robustly classify against an adversary which flips $O(1)$ bits. To see this, suppose we are given a point $x \in \{0,1\}^m$ drawn from $\mathcal{D}_0$ to classify. In order for any algorithm to correctly classify this point, we need to check that most within an λ-ball lie within the classifying set A. However, the number of points in $B(x,\lambda)$, where the λ-ball is now defined using the Hamming metric, grows exponentially with the number bit-flips, so efficiently checking whether most are contained in A is not possible unless $\lambda = O(1)$.

To increase the robustness of the classifier, we can use classical error correcting codes to first encode the output of the PRG as proposed in Reference [208]. This method allows us to robustly classify points with as many bit-flips as can be tolerated by the chosen error correcting code. As there exist error-correcting codes that can uniquely decode under the presence of $O(l)$ errors, where l is the length of the code words [220], this increases the robustness of the classification algorithm to $O(l)$ as well. If we denote the encoding and decoding functions of the error correcting code as

$$\text{Encode} : \{0,1\}^m \to \{0,1\}^{l(m)} \tag{4.5}$$

$$\text{Decode} : \{0,1\}^{l(m)} \to \{0,1\}^m, \tag{4.6}$$

respectively, where l is some polynomial, the classification task is now defined as in Construction 4.2.1 where the two distributions are given by the encoded PRG. The two distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ are now given by the distributions of

$$(0, \text{Encode}(G(X))) \text{ and } (1, \text{Encode}(Y)), \tag{4.7}$$

where once again X is drawn uniformly from $\{0,1\}^n$ and Y is drawn uniformly from $\{0,1\}^m$. The adversary can now flip up to $O(l(m))$ bits after which we apply the decoding function. We can then efficiently classify points if we can break the PRG G, as in the previous section. The converse is also true: if we can efficiently classify points, then we can break the PRG.

4.5 Necessity of a training phase when prior information is reduced

Although the previous constructions allow for a rigorous quantum advantage in terms of the computational complexity of outputting a correct label once the hypothesis is learned, when provided with a sample to classify the quantum algorithms from Sections 4.3 and 4.4 can output a correct label with high accuracy even if the algorithm was provided no previous samples. In other words, there is no training phase for the learning algorithm and the sample complexity to learn a hypothesis for the quantum algorithm is 0. This is inconsistent with the intuitive notion that learning requires a training phase, rather than an algorithm being able to simply calculate the correct output without learning from data. To remedy this problem, we can provide less information to the quantum algorithm before hand and instead supply this information to the algorithm during a training phase.

Breaking the PRG we use for our construction requires knowing both the prime p used as the modulus as well as the generator g of the PRG. Instead of assuming that these values are known beforehand, we no longer explicitly provide the algorithm with the generator g which is required to run Shor's algorithm. Instead, single bits of the pair g are appended to the data when we query for a sample.

Our training set now consists of samples of the form

$$(0, \text{Encode}(G(X)||Z||b_g(Z))) \text{ and } (1, \text{Encode}(Y||Z||b_g(Z)), \tag{4.8}$$

where the marginal distribution of Z is uniform on $\{1,\ldots,p\}$ and $b_g : \{1,\ldots,p\} \to \{0,1\}$ is a function which on input i returns the i'th bit of g. In the training phase, we receive polynomially many samples of the form Equation (4.8). As shown in Section 4.10, if p is an n-bit safe prime and we are given access to $n\log(n/\delta)$ samples, then with probability at least $1-\delta$ we have enough samples to perfectly reconstruct g. In this case, we can then use Shor's algorithm as in the previous sections in order to label points.

We note that it would be also possible to simply append all of g to each data point, rather than providing the information one bit at a time. However, if all of g is

appended to each data point, then once again no training phase is needed. In order to classify point without having seen training data, we first read off the bits describing g. After we have g, we can then classify points if given access to quantum resources. To avoid this problem, we provide g one bit at a time.

4.6 Quantum advantage for an adversarial classification problem

In the constructions above, we can use a PRG which is classically safe but can be broken efficiently using Shor's algorithm, and an error correcting code which can uniquely decode up to a linear number of errors. In the presence of an adversary which always sets the first bit to 0 and can also flip up to a linear number of bits, accurate classification implies the ability to distinguish between the outputs of the PRG and the uniform distribution. Because we choose a classically safe (under widely accepted cryptographic assumptions) PRG, the classification task must necessarily be inefficient. However, since the PRG is not quantum safe, we can show that classification can be done efficiently. We thus have a quantum advantage for an adversarial classification problem.

We summarize our result in the following theorem.

Theorem 4.6.1. *There exists a binary classification problem on $\mathbb{Z}_2^l$ equipped with the Hamming metric that satisfies:*

1. *Non-robust classification is classically sample and computationally efficient to learn and implement.*
2. *There exists a $\Theta(l)$-robust classical classifier.*
3. *There exists an $O(1)$-adversary such that classification is classically sample and computationally inefficient to learn.*
4. *There exists an $O(l)$-robust sample and computationally efficient quantum learner whose output is computationally efficient.*

To prove theorem 4.6.1, we first state a learning algorithm for binary classification, then show that each of the 4 properties of Theorem 4.6.1 are satisfied. The first part of the learning algorithm is the training phase. In the following, n is the length of an n-bit safe prime.

Algorithm 1 Training phase of the quantum PAC learning algorithm for λ-robust binary classification

procedure LEARN(δ, n)
  $N \leftarrow n \log(n/\delta)$ ▷ Set number of samples
  $g \leftarrow \text{list}(n)$ ▷ Initialize empty lists for storing bits of g
  for $iteration = 1, 2, \ldots, N$ **do**
    $x_0 \leftarrow \text{Sample}\left(\mathcal{D}_0^i\right)$ ▷ Receive a sample from each distribution
    $x_0 \leftarrow \text{Decode}(x_0)$ ▷ After decoding, x_0 is of the form $g^a||g^b||g^c||z||b_g(z)$
    $g[z] \leftarrow b_g(z)$
  end for
  return g ▷ The bitstring representing g
end procedure

The purpose of the training phase is to collect enough information about the generator g to run Shor's algorithm. As shown in Section 4.10, with probability at least $1 - \delta$ we can perfectly reconstruct g.

Once we have run the training phase, we can then output a quantum algorithm which can robustly classify points. The classification algorithm we output is as follows.

Algorithm 2 Algorithm for λ-robust binary classification

procedure CLASSIFY(x)
  $x \leftarrow \text{Sample}\left(\mathcal{D}_0^i, \mathcal{D}_1^i\right)$ ▷ Receive a sample from either distribution
  $x \leftarrow \text{Decode}(x)$ ▷ After decoding, x is of the form $g^a||g^b||g^c||z||b_g(z)$
  $a, b, c \leftarrow \text{SHOR}(g^a), \text{SHOR}(g^b), \text{SHOR}(g^c)$ ▷ Use Shor's algorithm to compute DL
  if $ab = c$ **then**
    return 0
  else
    return 1
  end if
end procedure

Construction of the PRG and error correcting code

We discuss here the PRG and error correcting code we use in order to construct the classification problem which satisfies all four requirements of Theorem 4.6.1. The construction relies on the DDH assumption over the group family of quadratic residues modulo a safe prime [219]. We use this particular construction because it is widely believed to be able to be used to construct a secure classical PRG which

is not quantum safe, as efficient implementation of the discrete logarithm allows us to efficiently invert the PRG. Efficient inversion of the PRG then allows us to efficiently determine whether a bitstring was sampled from the uniform distribution or the output of the PRG, which in turn allows for efficient classification.

A safe prime is a prime number p which can be written in the form $p = 2q + 1$, where q is also prime. The DDH assumption applies to group families, families of groups which are parametrized by some index set. For our construction, we use the group family defined by the group of quadratic residues modulo a safe prime. See Section 4.8 for definitions.

We next informally define the DDH assumption. The rigorous definition can be found in Section 4.8.

Definition 4.6.1 (Decisional Diffie-Hellman assumption, informal)**.** *Let I be an infinite index set. We say that a group family $G = \{G_i\}_{i \in I}$ satisfies the DDH assumption if for all polynomial time algorithms $\mathcal{A}$ and all polynomials q*

$$\left|\mathbb{P}\left[\mathcal{A}(g^a, g^b, g^{ab}) = 1\right] - \mathbb{P}\left[\mathcal{A}(g^a, g^b, g^c) = 1\right]\right| < \frac{1}{q(n)}$$

for sufficiently large n, where g is a group generator for G_i and $n = |i|$ is the length of the binary representation of p. The probability is taken over the randomness in the algorithm $\mathcal{A}$, as well as over the randomness in selecting a group and group generator g from the group family.

Roughly speaking, if the DDH assumption holds over a group family, then (g^a, g^b, g^{ab}) and (g^a, g^b, g^c) are computationally indistinguishable when a, b and c are drawn uniformly from the integers in the range $[0, |G_i| - 1]$.

We now define the PRG we use in our construction. We use the quadratic residues modulo p where p is a safe prime to construct a PRG. We represent each $h \in \mathrm{QR}_p$ by its binary representation with zero padding on the left if $|h| < |p|$. Then every binary representation of elements of QR_p has length $\lfloor \log p \rfloor + 1$. Given a group generator g, the PRG is defined as

$$G_g(a, b) = g^a || g^b || g^{ab},$$

where a and b are drawn independently and uniformly from $[0, q - 1]$, and $||$ represents concatenation. All exponentiation is done modulo p. By the DDH assumption, $G_g(a, b)$ is computationally indistinguishable from $g^a || g^b || g^c$, where

c is also drawn independently and uniformly from $[0, q-1]$. Since g is a generator of a cyclic group and a, b, c are drawn uniformly, $g^a||g^b||g^c$ is uniformly distributed over QR_p^3.

Although bit flip perturbations to $g^a||g^b||g^c$ do not necessarily map to another element of QR_p^3, since we will use an error correcting code on top of $g^a||g^b||g^c$, once decoded and with sufficiently few bit flips the original string will still be recovered.

We now specify the error correcting code we use in our construction. We use the fact that there exist classical error correcting codes which can correct for a number of errors in proportion to the length of the encoded string [220]. In particular, we use Theorem 10 from Ref. [220]:

Theorem (Binary code of rate $\Omega(\gamma^3)$ with relative distance $(1/2-\gamma)$)**.** *For any $\gamma < 1/4$ there exists a binary code $C : \{0,1\}^m \to \{0,1\}^l$ with $l = O(m/\gamma^3)$ which can detect and correct up to $(1/4-\gamma)l$ errors uniquely and which has an encoding time $O(l \log^{O(1)} l)$ and decoding time $O(l^2 + 2^{1/\gamma^3})$.*

Following [208], we define, for fixed γ, $\mathrm{Encode}_\gamma : \{0,1\}^m \to \{0,1\}^l$ as the encoding function and $\mathrm{Decode}_\gamma : \{0,1\}^l \to \{0,1\}^m$, which is well defined since the code supports unique decoding.

Informal proof of Theorem 4.6.1

Using the construction in Section 4.6, we outline the proof of Theorem 4.6.1, noting that we are particularly interested in parts 3 and 4 of the theorem. For simplicity, we fix $\gamma = 1/8$ for the error correction code and drop the subscript γ for the encoding and decoding functions. The distribution $\mathcal{D}_0$ is defined as being the uniform distribution over the set

$$D_0 = \left\{0||\mathrm{Encode}(g^a||g^b||g^{ab}||z||b_g(z)) \mid a, b \in \mathbb{Z}_q,\ g \in \mathrm{QR}_p,\ z \in \{1, \ldots, p\}\right\} \tag{4.9}$$

and $\mathcal{D}_1$ as uniform over the set

$$D_1 = \left\{1||\mathrm{Encode}(g^a||g^b||g^c||z||b_g(z)) \mid a, b, c \in \mathbb{Z}_q,\ g \in \mathrm{QR}_p,\ z \in \{1, \ldots, p\}\right\}. \tag{4.10}$$

Here, g is a generator of QR_p, g^x is to be understood as the left zero-padded binary representation of the integer, padded to length $\lfloor \log p \rfloor + 1$, and $||$ represents concatenation. Since the length of each g^x is $\lfloor \log p \rfloor + 1$, we have $m = 4(\lfloor \log p \rfloor +$

1) + 1 for the inputs into the encoding function and $l = O(m)$. The length of each element of D_0 and D_1 is therefore $l + 1$.

For the proof, we assume we have already trained on $n \log(n/\delta)$ samples, so that with probability $1 - \delta$ we have reconstructed g. The classical non-robust classifier outputs the first bit of the input, proving part 1.

For part 2, suppose we are given a bit string of length $l + 1$, of the form $x_0||\text{Encode}(g^a||g^b||g^c||z||b_g(z))$. We define the classifying set as

$$A = \cup_{x \in D_0} B(x, l/8). \tag{4.11}$$

As shown in Section 4.10, with the classifying set defined as above, the existence of a $\Theta(l)$-robust classifier depends only on the relative sizes of the sets D_0 and D_1. In particular, if $|D_0|/|D_1|$ is negligible, then a $\Theta(l)$-robust classifier exists. The number of elements in D_0 is $2pq^2$ and the number of elements in D_1 is $2pq^3$, so $|D_0|/|D_1| = 1/q$. Since q grows exponentially relative to the length of the bit string representation of p, the relative sizes of the sets is negligible and a $\Theta(l)$-robust classifier exists.

For part 3, we assume for contradiction that we have an efficient $O(1)$-robust classical classifier $\mathcal{A}$:

$$\left|\mathbb{P}_{X \sim \mathcal{D}_0}[\mathcal{A}(X) = 1] - \mathbb{P}_{X \sim \mathcal{D}_1}[\mathcal{A}(X) = 1]\right| < \frac{1}{n^\alpha}, \tag{4.12}$$

where $\alpha > 0$ is constant and n is the length of the input. The adversary that always erases and sets the first bit to 0, so we can ignore it. Since the remaining bits of X are either encoded Diffie-Hellman triples if $X \sim \mathcal{D}_0$ or encoded triples of all group elements if $X \sim \mathcal{D}_1$, this directly contradicts the DDH assumption since we can use $\mathcal{A}$ to efficiently distinguish between triples and non-triples.

For part 4, we receive a bit string $0||\text{Encode}(g^a||g^b||g^c||z||b_g(z))$ with up to $l/8 - 1$ bits flipped on the encoded suffix. We apply Decode to the suffix, yielding the string $g^a||g^b||g^c$. Using Shor's algorithm for discrete logarithms, we can find a, b and c in quantum polynomial time. We then output 0 if $ab = c$ and 1 if $ab \neq c$. Since both the decoding function and Shor's algorithm run in polynomial time, this classification is efficient.

4.7 Pseudorandom generators

In the following sections we provide the rigorous definitions required to rigorously prove quantum advantage for adversarially robust machine learning.

In this section we provide definitions and results without proof to rigorously define pseudorandom generators (PRGs). We assume that we are always referencing a specific probability space $(\Omega, \mathcal{F}, \mathbb{P})$, although we rarely need to make use of any specific details of the sample space or σ-algebra. For more details, see Ref. [221].

Pseudorandom generators are ensembles or families of functions whose outputs "appear random" when seeded with a uniformly (truly) random input. A probability ensemble is defined as

Definition 4.7.1 (Probability ensemble)**.** *Let I be an index set. A probability ensemble indexed by I is a sequence $\{X_i\}_{i\in I}$ of random variables where each $X_i : \Omega \to \mathbb{R}$ is a random variable.*

As we are interested in asymptotic results, notions such as two ensembles being "similar" or functions being "hard to invert" are defined in terms of so-called negligible functions, which captures the notion of rapidly decreasing functions as the length of the inputs increase. A negligible function is defined as

Definition 4.7.2 (Negligible functions)**.** *A function $\mu : \mathbb{N} \to \mathbb{R}$ is negligible if $\mu(n) < 1/p(n)$ for all polynomials p and all sufficiently large n.*

We now define what it means for two ensembles to be indistinguishable. There are two related notions of indistinguishability. The classical notion of indistinguishability is statistical closeness, defined in terms of the statistical difference, also known as the total variation distance

$$\Delta(n) = \frac{1}{2}\sum_{\alpha} \left|\mathbb{P}[X_n = \alpha] - \mathbb{P}[Y_n = \alpha]\right|. \tag{4.13}$$

Two ensembles are said to be statistically close if their statistical difference is negligible. Statistical closeness is a stronger than necessary requirement, since for the purposes of complexity theory it does not matter if two ensembles are statistically close, only whether there exists an efficient algorithm which can distinguish between the two ensembles. This notion is captured by computational indistinguishability:

Definition 4.7.3 (Polynomial-time computational indistinguishability)**.** *Two ensembles $\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$ are polynomial-time computationally indistinguishable if for every probabilistic polynomial-time algorithm D*

$$\left|\mathbb{P}[D(X_n) = 1] - \mathbb{P}[D(Y_n) = 1]\right| \tag{4.14}$$

is a negligible function of n.

Two ensembles which are polynomial-time indistinguishable may not be statistically close, but no efficient algorithm can detect the difference between them.

PRGs are used to "expand randomness," in the sense that they take as input a uniformly chosen random seed and output a longer bit-string which "appears random." Formally, we first define pseudorandom ensembles:

Definition 4.7.4 (Pseudorandom ensembles)**.** *An ensemble $\{X_n\}_{n\in\mathbb{N}}$ is called pseudorandom if it is polynomial-time computationally indistinguishable from some uniform ensemble $\{U_{l(n)}\}_{n\in\mathbb{N}}$, where l is a polynomial.*

Pseudorandom ensembles can be used in any polynomial time application which requires uniform randomness with negligible degradation in performance. Provided that any adversary must run in polynomial time, the probability of detecting that a pseudorandom ensemble was used rather than a uniformly random ensemble is negligible.

We can now formally define the concept of pseudorandom generators.

Definition 4.7.5 (Pseudorandom generators)**.** *A pseudorandom generator G is a deterministic polynomial-time algorithm which satisfies the following conditions:*

1. *Expansion: there exists a function $l : \mathbb{N} \to \mathbb{N}$ such that $l(n) > n$ for all $n \in \mathbb{N}$, and $|G(s)| = l(|s|)$ for all $s \in \{0,1\}^*$.*
2. *Pseudorandomness: the ensemble $\{G(U_n)\}_{n\in\mathbb{N}}$ is pseudorandom.*

The polynomial l in the above definition is called the expansion factor of the PRG and the input s is called its seed. Since the ensemble $\{G(U_n)\}_{n\in\mathbb{N}}$ is pseudorandom, it is polynomial-time computationally indistinguishable from the uniform distribution $\{U_{l(n)}\}_{n\in\mathbb{N}}$. Provided that an adversary only has access to polynomial-time resources, we need only have access to uniform ensembles $\{U_n\}_{n\in\mathbb{N}}$ in order to build safe applications which require longer random strings.

PRGs used in practice rely on cryptographic hardness assumptions, such as the Decisional Diffie-Hellman over certain group families. These cryptographic assumptions are based on the hardness of inverting one-way functions, functions which are easy to compute in the forward direction but hard to invert. Finally, we note that in many modern applications, the underlying cryptographic assumptions are widely

accepted, yet with access to polynomial-time quantum resources are easily broken. The use of so-called "quantum-safe" cryptographic assumptions are an active area of research [32, 222].

4.8 The Decisional Diffie-Hellman assumption

The Decisional Diffie-Hellman (DDH) assumption underlies many modern cryptosystems [223, 224, 225]. For more details, see Ref. [219] and [198] which this section is based on.. In this section, we again assume that we are always referencing a specific probability space $(\Omega, \mathcal{F}, \mathbb{P})$. For a fixed cyclic, multiplicative group G with group generator g, and uniformly random integers $a, b, c \in [1, |G|]$, the DDH assumption states that, roughly speaking, (g^a, g^b, g^{ab}) and (g^a, g^b, g^c) are polynomial-time computationally indistinguishable. We give two examples of group for which the DDH assumption is believed to hold:

1. Let $p = 2q + 1$ where both p and q are prime, and let $G = \mathrm{QR}_p$ be the group of quadratic residues modulo p.

2. Let $N = pq$, where all of $p, q, (p-1)/2$ and $(q-1)/2$. Let G be the cyclic subgroup of order $(p-1)(q-1)$.

We note that in the above, both groups are parametrized by a single parameter, p or N. Some groups for which the DDH assumption is believed to hold may be parametrized by multiple parameters, but we always define the DDH assumption over parameterized group families. Computational indistinguishability will be defined in terms of the parameters of the group families, allowing for asymptotic analysis as the length of the binary representation of the parameter increases.

We next formally define the DDH assumption, for which we need several preliminary definitions. First, since the DDH assumption is an assumption on group families, we formally define efficient group families.

Definition 4.8.1 (Efficient group families)**.** *An efficient group family is a sequence of cyclic groups $G = \{G_p\}_{p \in P}$ where P is an infinite index set. We denote by $|p|$ the length of the binary representation of p. By efficient, we mean that there exists a polynomial time (in $|p|$) algorithm which, when given p and two elements of G_p outputs their product. We denote by $\mathcal{ID}$ the instance description, a probabilistic polynomial-time algorithm that on input p outputs a group generator for G_p.*

Since the DDH assumption is based on group families, we need an efficient way to select groups from the family given some integer input. We do this by defining instance generators:

Definition 4.8.2 (Instance generator)**.** *An instance generator $\mathcal{IG}$ for G is a probabilistic polynomial-time algorithm which on unary input n outputs an index $p \in P$ and a group generator $g \in G_p$. We note that for each n, $\mathcal{IG}$ induces a distribution on the set of indices $p \in P$ and that the output $|p|$ is polynomial in n.*

We can now formally define the DDH assumption.

Definition 4.8.3 (Decisional Diffie-Hellman assumption)**.** *Let $G = \{G_p\}_{p \in P}$ be a group family with instance description $\mathcal{ID}$ and instance generator $\mathcal{IG}$. We say that the DDH assumption holds over G if for all probabilistic polynomial-time algorithms $\mathcal{A}$,*

$$\left|\mathbb{P}_{\substack{p \leftarrow \mathcal{IG}(1^n) \\ g \leftarrow \mathcal{ID}(p) \\ a,b \leftarrow \mathbb{Z}_{|G_p|}}}[\mathcal{A}(p, g, g^a, g^b, g^{ab}) = 1] - \mathbb{P}_{\substack{p \leftarrow \mathcal{IG}(1^n) \\ g \leftarrow \mathcal{ID}(p) \\ a,b,c \leftarrow \mathbb{Z}_{|G_p|}}}[\mathcal{A}(p, g, g^a, g^b, g^c) = 1]\right| \tag{4.15}$$

is a negligible function of n. By $p \leftarrow \mathcal{IG}(1^n)$ ($g \leftarrow \mathcal{ID}(p)$) we mean that p (g) is distributed according to the distribution induced by the randomness of the algorithm $\mathcal{IG}$ ($\mathcal{ID}$). By $a, b, (c) \leftarrow \mathbb{Z}_{|G_p|}$, we mean that $a, b, (c)$ are independently distributed according to the uniform distribution on $\mathbb{Z}_{|G_p|}$.

We now focus on the DDH assumption on the group family defined by the quadratic residues modulo a safe prime. The group of quadratic residues modulo an integer is defined as

Definition 4.8.4 (Group of quadratic residues over $\mathbb{Z}_N^*$)**.** *Let $\mathbb{Z}_N^*$ denote the multiplicative group of integers modulo N. An element $y \in \mathbb{Z}_N^*$ is a quadratic residue modulo N if $y \equiv x^2 \pmod{N}$ for some $x \in \mathbb{Z}_N^*$. We denote by QR_N the subgroup of quadratic residues modulo N.*

A safe prime is a prime p of the form $p = 2q + 1$, where q is also prime and the parametrization set P is the set of all safe primes. We note that $\{\mathrm{QR}_p\}_{p \in P}$ is indeed a valid efficient group family, since computing the product of two elements in QR_p modulo p is efficient [226] and an instance description can, for example, be constructed by randomly selecting an integer from $[1, p]$, followed by (efficient)

membership testing using Euler's criterion [227]. Since exactly half of the elements in $\mathbb{Z}_p^*$ are quadratic residues ($|\mathrm{QR}_p| = q$), and all non-identity elements of QR_p are generators, this method will rapidly find a generator for QR_p.

Finally, we show how to construct a PRG which is provably pseudorandom under the DDH assumption over the quadratic residues modulo a safe prime. We define the parametrized function $G_{g,p} : \mathbb{Z}_q \times \mathbb{Z}_q \to \mathrm{QR}_p \times \mathrm{QR}_p \times \mathrm{QR}_p$ by

$$G_{g,p}(a, b) = g^a || g^b || g^{ab} \tag{4.16}$$

where all exponentiation is done modulo p (efficiently, see Ref. [228]) and $||$ represents concatenation. This family of functions is a pseudorandom generator under the DDH assumption over QR_p, as we can see by rewriting Equation (4.15) as

$$\left|\mathbb{P}_{\substack{p\leftarrow\mathcal{IG}(1^n)\\ g\leftarrow\mathcal{ID}(p)\\ a,b\leftarrow\mathbb{Z}_{|G_p|}}}[\mathcal{A}(p, g, G_{g,p}(a, b)) = 1] - \mathbb{P}_{\substack{p\leftarrow\mathcal{IG}(1^n)\\ g\leftarrow\mathcal{ID}(p)\\ a,b,c\leftarrow\mathbb{Z}_{|G_p|}}}[\mathcal{A}(p, g, g^a, g^b, g^c) = 1]\right|. \tag{4.17}$$

Since the quantity in Equation (4.17) is negligible for all probabilistic polynomial-time algorithms $\mathcal{A}$, this implies that no efficient algorithm can distinguish between $G_{g,p}(a, b) = g^a || g^b || g^{ab}$ and $g^a || g^b || g^c$ with non-negligible probability. Since $g^a || g^b || g^c$ is uniformly distributed over QR_p^3 (for uniformly distributed $a, b, c \in \mathbb{Z}_q$), both conditions of Definition 4.7.5 are satisfied and $G_{g,p}$ is a PRG.

4.9 Volume arguments for the existence of good classifiers

Here we present the volume arguments used to show that when $\mathcal{D}_0$ is the distribution of the output of a PRG and $\mathcal{D}_1$ is the uniform distribution, non-robust classification is computationally and sample efficient and that robust classification is possible but inefficient. First we present the volume argument for the non-robust case.

Volume argument for non-robust classification

We denote by $G : \{0, 1\}^n \to \{0, 1\}^m$ the PRG, and we define $\mathcal{D}_0$ to be the distribution of $G(x)$ for x uniform in $\{0, 1\}^n$, and $\mathcal{D}_1$ the uniform distribution over $\{0, 1\}^m$. We define A to be the range of G, $A = \{x \in \{0, 1\}^m : x = G(y) \text{ for some } y \in \{0, 1\}^n\}$.

Then $\mathbb{P}_{X\sim\mathcal{D}_0}[X \in A] = 1$ and

$$\mathbb{P}_{X\sim\mathcal{D}_1}[X \notin A] = \frac{1}{2^m}\sum_{x=0}^{2^m-1} \mathbf{1}_{[x\notin \mathrm{Im}(G)]} \tag{4.18}$$

$$\geq \frac{1}{2^m}(2^m - 2^n) \tag{4.19}$$

$$= 1 - 2^{n-m} \tag{4.20}$$

since $|\mathrm{Range}(G)| \leq 2^n$. $1 - 2^{n-m} \geq 0.99$ for large enough n provided that m is an increasing function of n, which is the case in all of our arguments.

Volume argument for robust classification

In this section, we provide a volume argument to show that $\Theta(\sqrt{n})$ robust classification is possible yet inefficient if we do not use error correcting codes. Similarly to the non-robust case, we define A as

$$A = \cup_{x\in\mathrm{Im}(G)} B(x, \epsilon) \tag{4.21}$$

and once again we have $\mathbb{P}_{X\sim\mathcal{D}_0}[X \subset A] = 1$.

For $\mathcal{D}_1$, we show that

$$\mathbb{P}_{X\sim\mathcal{D}_1}[B(X, \epsilon) \cap A \neq \varnothing] \leq 0.01 \tag{4.22}$$

by counting the total number of points within a 2ϵ-ball of all points in $\mathrm{Im}(G)$.

$$\mathbb{P}_{X\sim\mathcal{D}_1}[B(X, \epsilon) \cap A \neq \varnothing] = \frac{1}{2^m}\sum_{x=0}^{2^m-1} \mathbf{1}_{[B(x,\epsilon)\cap A\neq\varnothing]} \tag{4.23}$$

$$= \frac{1}{2^m}\left[\sum_{x\in\mathrm{Im}(G)} \mathbf{1}_{[B(x,\epsilon)\cap A\neq\varnothing]} + \sum_{x\notin\mathrm{Im}(G)} \mathbf{1}_{[B(x,\epsilon)\cap A\neq\varnothing]}\right] \tag{4.24}$$

$$\leq \frac{1}{2^m}\left[2^n + \sum_{x\notin\mathrm{Im}(G)} \mathbf{1}_{[B(x,\epsilon)\cap A\neq\varnothing]}\right] \tag{4.25}$$

$$= 2^{n-m} + \frac{1}{2^m}\sum_{x\notin\mathrm{Im}(G)} \mathbf{1}_{[B(x,\epsilon)\cap A\neq\varnothing]}. \tag{4.26}$$

To complete the argument, we need to upper bound

$$\sum_{x\notin\mathrm{Im}(G)} \mathbf{1}_{[B(x,\epsilon)\cap A\neq\varnothing]}.$$

To do this, we consider any point $x \notin \mathrm{Im}(G)$. Since $B(x,\epsilon) \cap A \neq 0$ iff there exists a point $y \in \mathrm{Im}(G)$ within 2ϵ of x, we have that $\sum_{x \notin \mathrm{Im}(G)} \mathbf{1}_{[B(x,\epsilon)\cap A \neq \emptyset]}$ is equal to the number of points in $\{0,1\}^m \backslash \mathrm{Im}(G)$ that are within 2ϵ of a point in ImG. This must be less than the total number of points within 2ϵ of all points in $\mathrm{Im}(G)$. In other words,

$$\sum_{x \notin \mathrm{Im}(G)} \mathbf{1}_{[B(x,\epsilon)\cap A \neq \emptyset]} \leq |\mathrm{Im}(G)|M,$$

where M is the number of points in $\{0,1\}^m$ inside of a 2ϵ-ball centered at any point, since each ball contains the same number of points. In the following, we also assume that $m = kn$ for some constant $k > 1$. For the PRG we use, we have that $k > 3/2$. We thus need to count how many points are inside this ball. By symmetry, we can take the center of the ball to be 0^m and count how many bit-strings are inside this ball. We do this by writing the Hamming weight as $H(x) = \sum_{i=1}^m x_i = \sum_{i=1}^m x_i^2 = |x|_2^2$ since x is binary. For a ball of size 2ϵ (in L_2 norm), we then have that the number of strings in the 2ϵ-ball is

$$\begin{aligned}
M &= \sum_{i=0}^{\lfloor 4\epsilon^2 \rfloor} \binom{m}{i} && (4.27)\\
&= \sum_{i=0}^{d} \binom{m}{i} \quad \text{setting } d = \lfloor 4\epsilon^2 \rfloor && (4.28)\\
&\leq \left(\frac{em}{d}\right)^d && (4.29)\\
&= 2^{nc \log \frac{ke}{c}} \quad \text{setting } d = cn. && (4.30)
\end{aligned}$$

Then,

$$\begin{aligned}
\sum_{x \notin \mathrm{Im}(G)} \mathbf{1}_{[B(x,\epsilon)\cap A \neq \emptyset]} &\leq |\mathrm{Im}(G)|M && (4.31)\\
&\leq 2^n 2^{nc \log \frac{ke}{c}} && (4.32)\\
&= 2^{n(c \log \frac{ke}{c} + 1)}. && (4.33)
\end{aligned}$$

Substituting back into Equation (4.26), we get

$$\begin{aligned}
\mathbb{P}_{X \sim \mathcal{D}_1}\left[B(X,\epsilon) \cap A \neq \emptyset\right] &\leq 2^{n-m} + \frac{1}{2^m} \sum_{x \notin \mathrm{Im}(G)} \mathbf{1}_{[B(x,\epsilon)\cap A \neq \emptyset]} && (4.34)\\
&\leq 2^{n-m} + 2^{-m} 2^{n(c \log \frac{ke}{c} + 1)} && (4.35)\\
&= 2^{n-m} + 2^{n(c \log \frac{ke}{c} + 1 - k)}. && (4.36)
\end{aligned}$$

Provided that $c \log \frac{ke}{c} + 1 - k$ is negative, we have our desired result by taking n sufficiently large. The function $f(x) = x \log \frac{ke}{x} + 1 - k$ is negative for sufficiently small x for all $k > 1$. Since d is the number of bit flips we allow, i.e., the maximum allowable Hamming distance, and $\epsilon = O(\sqrt{d})$, we have that $O(\sqrt{m})$-robust classification is possible. However, since the adversary always flips the first bit, the classification must be necessarily be inefficient otherwise we violate the assumption of the PRG being computationally indistinguishable from the uniform distribution.

4.10 Rigorous proof of Theorem 4.6.1

Here we provide a rigorous proof of Theorem 4.6.1.

Sample efficiency

We first prove that for $n = \lfloor \log p \rfloor + 1$, with p an n-bit safe prime, that with probability $1 - \delta$ we can perfectly reconstruct a generator g for QR_p given access to $n \log (n/\delta)$ samples of the form $x_0 || G_{g,p}(a, b) || z || b_g(z)$.

With $k = n \log (n/\delta)$ i.i.d. training samples, drawn from the uniform distributions over either $D_{0,g}$ or $D_{1,g}$, we collect samples of the suffixes

$$(z_1, b_g(z_1)), (z_2, b_g(z_2)) \ldots (z_k, b_g(z_k)). \tag{4.37}$$

It is not important how we select which distribution to draw from, since the suffixes are uniformly random in either case. Each sample $(z_1, b_g(z_1))$ gives us 1 bit of information about g. If we have enough samples to learn all of g, then we have found the correct concept and will be able to classify 100% accurately. To see how many samples we need to draw, we write the binary expansion

$$g = g_{n-1} g_{n-1} \cdots g_0. \tag{4.38}$$

We want to know given k samples, what is the probability of seeing each g_i at least once. Let's call this event A. If we have less than n samples, then clearly the probability of seeing each bit at least once is 0. For $k \geq n$ samples, let us define the events

$$A_i = \text{ the event that we see bit } i \text{ at least once.} \tag{4.39}$$

Then, for $m \geq n$,

$$\begin{aligned}
\mathbb{P}[A] &= \mathbb{P}[A_0 \wedge A_1 \wedge \cdots \wedge A_{n-1}] && (4.40)\\
&= 1 - \mathbb{P}[A_0^c \vee A_1^c \vee \cdots \vee A_{n-1}^c] && (4.41)\\
&\geq 1 - \sum_{i=0}^{n-1} \mathbb{P}[A_i^c] \quad \text{union bound} && (4.42)\\
&= 1 - n\mathbb{P}[A_0^c] \quad \text{independence} && (4.43)\\
&= 1 - n\left(\frac{n-1}{n}\right)^m && (4.44)\\
&\geq 1 - n\exp\left(-\frac{m}{n}\right). && (4.45)
\end{aligned}$$

Setting

$$\delta = n\exp\left(-\frac{m}{n}\right) \tag{4.46}$$

we have that

$$\begin{aligned}
\frac{\delta}{n} &= \exp\left(-\frac{m}{n}\right) && (4.47)\\
\implies \frac{m}{n} &= \log\frac{n}{\delta} && (4.48)\\
\implies m &= n\log\frac{n}{\delta}. && (4.49)
\end{aligned}$$

In summary, we have that with probability at least $1-\delta$, we can perfectly reconstruct g provided that we have at least $n\log(n/\delta)$ training samples.

Proof of Theorem 4.6.1 assuming g has been reconstructed

Here, we assume that we have run the training phase of the algorithm and have reconstructed g. We omit the last bits of the data $z||b_g(z)$ for notational simplicity.

As in Section 4.6 of the main text, we set $\gamma = 1/8$ for our error correcting code with encoding and decoding functions $\text{Encode} : \{0,1\}^m \to \{0,1\}^l$ and $\text{Decode} : \{0,1\}^l \to \{0,1\}^m$, respectively. We define the PRG as in Equation (4.16), $G_{g,p}(a,b) = g^a||g^b||g^{ab}$ with $p = 2q+1$, p, q prime, and $a, b \in \mathbb{Z}_q$. We denote by $|x|$ the length of the binary representation of x if x is a numerical value and the number of elements in x if x is a set. The adversary is chosen such that it always sets the first bit to zero, then can flip up to $l/8$ bits arbitrarily. The distribution $\mathcal{D}_0$ is defined as being the uniform distribution over the set

$$D_0 = \left\{0||\text{Encode}(G_{g,p}(a,b)) \mid a, b \in \mathbb{Z}_q,\ g \in \text{QR}_p\right\} \tag{4.50}$$

and $\mathcal{D}_1$ as uniform over the set

$$D_1 = \left\{1||\text{Encode}(g^a||g^b||g^c) \mid a, b, c \in \mathbb{Z}_q,\ g \in \text{QR}_p\right\}. \tag{4.51}$$

We define the classifying set, used for parts 1 and 2 of the proof, as

$$A = \cup_{x \in D_0} B(x, l/8). \tag{4.52}$$

Given a point $x \in \{0,1\}^{l+1}$, we label it 0 if $x \in A$ and 1 if $x \notin A$.

For part 1, non-robust classification, on input $x_0||x$ with $x_0 \in \{0,1\}$ and $x \in \{0,1\}^l$ we output the first bit x_0. Since $D_0 \subseteq A$, points from D_0 will always be correctly classified. However, since there is a non-zero overlap between the two sets D_0 and D_1 (when $ab = c$), there will always be a non-zero probability of misclassification. The classification error occurs when we output 1 for an element from D_1 which is in fact a Diffie-Hellman triple and so should be labeled 0. The probability of misclassification for points from D_1 is therefore upper bounded by $q^2/q^3 = 1/q$, since there are q^2 values for which $ab = c$ and q^3 elements in D_1. Since q grows exponentially in $|q|$ (and in $|p|$), the probability of misclassification is negligible.

For part 2, we need to show that the two conditions

$$\mathbb{P}_{X \sim \mathcal{D}_0}[B(X, \epsilon) \subset A] \geq 0.99 \quad \text{and} \quad \mathbb{P}_{X \sim \mathcal{D}_1}[B(X, \epsilon) \cap A = \varnothing] \geq 0.99, \tag{4.53}$$

are satisfied for $\epsilon < l/8$. The first inequality is immediate from the definition of A. For the second inequality, we again suppose we receive a bitstring $x_0||x$ with $x_0 \in \{0,1\}$ and $x \in \{0,1\}^l$. Since the adversary always sets the first bit to 0, we can obtain no useful information from that bit and so discard it. For the remaining bitstring x, we know that the adversary has flipped at most $l/8$ bits of x. We can therefore apply Decode to x to recover the unique string $g^a||g^b||g^c$. The classification can now be done in the decoded space, where we define

$$A' = \cup_{a,b \in \mathbb{Z}_q} g^a||g^b||g^{ab}. \tag{4.54}$$

Since the decoding is unique when fewer than $l/8$ bits are flipped, we have that

$$g^a||g^b||g^c \in A' \iff \text{Encode}(g^a||g^b||g^c) \in A. \tag{4.55}$$

Since in the decoded space we have no more adversarial perturbations, the misclassification error is again bounded by $D_0/D_1 = 1/q$, which is a negligible function of the length of the binary representation of p.

The proof of part 3 follows directly from Equation (4.17). We suppose that the first bit (and up to $l/8$ of the remaining bits) is flipped. We apply the Decode function. Then, an algorithm which can efficiently classify the decoded strings as either being $g^a||g^b||g^c$ or $g^a||g^b||g^{ab}$ would violate the DDH assumption for the PRG $G_{g,p}(a,b)$, and so an efficient classification algorithm cannot exist.

For part 4, we receive a bit string $0||\text{Encode}(g^a||g^b||g^c)$ with up to $l/8-1$ bits flipped on the encoded suffix. We apply Decode to the suffix, yielding the string $g^a||g^b||g^c$. Using Shor's algorithm for discrete logarithms, we can find a, b and c in quantum polynomial time. We then output 0 if $ab = c$ and 1 if $ab \neq c$. Since both the decoding function and Shor's algorithm run in polynomial time, this classification is efficient.

4.11 Discussion

Parts 3 and 4 of Theorem 4.6.1 provides the rigorous quantum advantage for adversarial machine learning. Although maximally robust classifiers exists for our construction (Theorem 4.6.1, part 2), they must necessarily be inefficient under the DDH assumption, and even $O(1)$-robust classical classifiers are inefficient. The quantum classifier is efficient in both sample complexity and computational complexity, since no explicit training examples are needed and once we receive a sample to classify, classification is efficient due to the efficiency of Shor's algorithm. On the other hand, under the DDH assumption, any classical algorithm is necessarily inefficient in regards to both sample and computational complexity.

We can view our results as a necessary condition of the use of quantum computers in real world adversarial machine learning tasks, in that if even for highly fine-tuned problems there exists no quantum advantage, then it may be unreasonable to expect any quantum advantage for real world problems with less mathematical structure. For there to exist a rigorous quantum advantage for real world problems, there must exist a rigorous quantum advantage for quantum-tailored problems. This illustrates one of the current difficulties in showing rigorous quantum advantages in real world machine learning tasks over classical learning algorithms.

This work establishes the first known rigorous quantum advantage for an adversarial machine learning task. Extensions to real space classification and real world adversarial machine learning tasks are active areas of investigation and will elucidate the exact quantum advantages to be expected.

Chapter 5

SUMMARY

In this thesis we have explored several applications of quantum computing. These applications range from quantum simulation algorithms on noisy devices to complexity theoretic proofs of the utility of quantum computers for machine learning tasks. Within each topic, we developed new machinery in order to tackle the specific technical challenges unique to the topic. The results obtained in this thesis have spurred new research directions for scientific applications of quantum computing in both the near and long term. We summarize here the main results obtained and conclude with an outlook of the prospects of future research related to the works of this thesis.

5.1 Quantum simulation algorithms for open quantum systems

In this project, we developed two new algorithms for simulating open quantum systems.

Algorithm I utilizes Quantum Imaginary Time Evolution (QITE) to implement the non-unitary evolution introduced when the density operator is vectorized. Vectorization of the density operator and the Lindblad equation results in a Schrödinger type equation with a non-Hermitian Hamiltonian. For an n qubit system, this algorithm requires $2n + 1$ qubits to simulate on quantum hardware. The time evolution generated by the non-Hermitian Hamiltonian results in non-unitary evolution. The unitary parts of the time evolution are implemented using standard quantum simulation techniques while the non-unitary parts are implemented using QITE. We also discussed the computational overheads, run-time bounds, and error analysis associated with the algorithm.

Algorithm II uses a purification based isomorphism to map a density operator to two copies of a pure state. Each copy can be propagated independently, resulting in a time evolution of an n qubit system requiring no ancilla qubits. Similar to Algorithm I, Algorithm II is analyzed in terms of its computational overheads, run-time bounds, and error characteristics.

Both algorithms were tested classically and on IBM Quantum hardware, where we simulated the spontaneous emission of a two-level system and the dissipative transverse field Ising model (TFIM).

5.2 Experimental realization of scalable probes of measurement induced phase transitions

In this study, we demonstrated an experimental realization of measurement-induced phase transitions (MIPTs) on superconducting quantum hardware using a cross-entropy protocol on up to 22 qubits. Compared to previous demonstrations of MIPTs on superconducting hardware, we were able to access larger systems while reducing the quantum resources required by two orders of magnitude.

The cross entropy protocol requires executing pairs of quantum circuits, one classically hard to simulate and one classically easy to simulate. The classically hard circuit requires the use of a quantum computer in order to efficiently probe the MIPT.

Due to the excessive circuit depths required for the larger systems, we used a Clifford based circuit compression technique which allowed us to halve the number of required qubits as well as significantly reduce the number of measurements required in a circuit. We improved on existing Clifford circuit compression methods by removing the requirement for adaptive quantum circuits, which was previously required in the best known circuit compression techniques.

We obtained results for 1D systems as well as infinite dimensional systems on up to 22 physical qubits, allowing us to probe MIPTs on system sizes of up to 44 sites. We also provided a method to use this protocol to benchmark quantum hardware with quantum circuits which use mid-circuit measurements.

Error mitigation techniques such as dynamical decoupling and readout error mitigation were tested to address noise in experimental setups but were found to have little impact. We also classically simulated noisy and noise-free circuits for both 1D and all-to-all systems. Our experimental results were consistent with both the noisy numerical simulations as well as theoretical predictions of critical phenomena.

5.3 Quantum advantage in adversarially robust machine learning

Chapter 4 explored the application of quantum computing to machine learning, particularly focusing on adversarial attacks where malicious interference with data is a concern. While extensive efforts have been dedicated to addressing adversarial robustness in classical machine learning, less attention has been given to its

quantum counterpart. We investigated whether quantum learning algorithms hold advantages in adversarial robustness compared to classical ones. By constructing an adversarial learning task that is hard for classical learners but easy for quantum ones, we demonstrated rigorously the potential of quantum computing in enhancing robustness in machine learning tasks.

5.4 Broader implications and future directions

The rapid progress in quantum hardware over the last several years has spurred much excitement on the prospects of using quantum computers to solve problems which are classically intractable. In this thesis, we have presented several applications which suggest that quantum speed-up may be possible, particularly for simulating open quantum systems and probing measurement induced phase transitions on noisy hardware. We have also provided a rigorous argument that fault tolerant quantum computers may also provide advantages for adversarially robust machine learning, albeit on a problem with no known applications. All of these topics are potentially of interest to the broader quantum computing community as well as the scientific community in general.

Quantum simulation algorithms for open quantum systems continues to be investigated by the community. Our work in this area could be extended by either reducing the ancilla qubit overhead or by finding alternative simulation algorithms which make use of alternative quantum resources such as mid-circuit measurements.

Measurement induced phase transitions have also garnered recent interest in the physics community as a phenomena which exhibits the universality of phase transitions in a purely quantum resource, entanglement. The work presented on MIPTs in this thesis could be further improved by reducing the computational complexity of the Clifford compression algorithm, which was the main bottleneck of our study, as well as by further investigating the effects of non-Pauli noise on the cross entropy benchmark that we examined.

Adversarially robust machine learning has many applications in real world settings, and so has resulted in the development of many attacks and defenses which can be implemented in real world applications. Rigorous quantum advantage in this setting was previously lacking, and our work helps to fill this knowledge gap. This work could be extended by providing a sufficient condition on when there exists a rigorous quantum advantage in adversarially robust machine learning, as our work can be viewed as a necessary condition for such an advantage. Additionally, finding

rigorous quantum advantages for more realistic problems would further motivate the use of quantum computing for machine learning tasks.

BIBLIOGRAPHY

[1] Paul Benioff. The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. In: *Journal of Statistical Physics* 22 (May 1980), pp. 563–591. URL: https://doi.org/10.1007/BF01011339.

[2] Richard P. Feynman. Simulating physics with computers. In: *International Journal of Theoretical Physics* 21.6/7 (1982), p. 22.

[3] Yuri Manin. Computable and uncomputable. Vol. 128. Moscow: Sovetskoye Radio, 1980.

[4] David Deutsch. Quantum theory, the Church–Turing principle and the universal quantum computer. In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400.1818 (1985), pp. 97–117. URL: https://doi.org/10.1098/rspa.1985.0070.

[5] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. In: *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*. 1993, pp. 11–20. URL: doi.org/10.1145/167088.167097.

[6] David Deutsch. Quantum computational networks. In: *Proceedings of the royal society of London. A. mathematical and physical sciences* 425.1868 (1989), pp. 73–90. URL: https://doi.org/10.1098/rspa.1989.0099.

[7] Chi-Chih Yao. Quantum circuit complexity. In: *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*. IEEE. 1993, pp. 352–361. URL: https://doi.org/10.1109/SFCS.1993.366852.

[8] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439.1907 (1992), pp. 553–558. URL: https://doi.org/10.1098/rspa.1992.0167.

[9] Don Coppersmith. An approximate Fourier transform useful in quantum factoring. In: *arXiv preprint quant-ph/0201067* (1994). URL: https://arxiv.org/abs/quant-ph/0201067.

[10] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In: *Proceedings 35th annual symposium on foundations of computer science*. Ieee. 1994, pp. 124–134. URL: https://doi.org/10.1109/SFCS.1994.365700.

[11] Alexei Y. Kitaev. Quantum measurements and the Abelian stabilizer problem. In: *arXiv preprint quant-ph/9511026* (1995). URL: https://arxiv.org/abs/quant-ph/9511026.

[12] Lov K. Grover. A fast quantum mechanical algorithm for database search. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996, pp. 212–219. URL: doi.org/10.1145/237814.237866.

[13] Seth Lloyd. Universal quantum simulators. In: *Science* 273.5278 (Aug. 1996), pp. 1073–1078. DOI: 10.1126/science.273.5278.1073. URL: doi.org/10.1126/science.273.5278.1073.

[14] Daniel S. Abrams and Seth Lloyd. Simulation of many-body Fermi systems on a universal quantum computer. In: *Physical Review Letters* 79.13 (1997), p. 2586. URL: https://doi.org/10.1103/PhysRevLett.79.2586.

[15] Christof Zalka. Simulating quantum systems on a quantum computer. In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 454.1969 (1998), pp. 313–322. URL: https://doi.org/10.1098/rspa.1998.0162.

[16] Bruce M. Boghosian and Washington Taylor IV. Quantum lattice-gas model for the many-particle Schrödinger equation in d dimensions. In: *Physical Review E* 57.1 (1998), p. 54. URL: https://doi.org/10.1103/PhysRevE.57.54.

[17] David G. Cory, Amr F. Fahmy, and Timothy F. Havel. Ensemble quantum computing by NMR spectroscopy. In: *Proceedings of the National Academy of Sciences* 94.5 (1997), pp. 1634–1639. URL: https://doi.org/10.1073/pnas.94.5.1634.

[18] Neil A. Gershenfeld and Isaac L. Chuang. Bulk spin-resonance quantum computation. In: *Science* 275.5298 (1997), pp. 350–356. URL: https://doi.org/10.1126/science.275.5298.350.

[19] Juan I. Cirac and Peter Zoller. Quantum computations with cold trapped ions. In: *Physical Review Letters* 74.20 (1995), p. 4091. URL: https://doi.org/10.1103/PhysRevLett.74.4091.

[20] Daniel Loss and David P. DiVincenzo. Quantum computation with quantum dots. In: *Physical Review A* 57.1 (1998), p. 120. URL: https://doi.org/10.1103/PhysRevA.57.120.

[21] Yasunobu Nakamura, Yu A. Pashkin, and John S. Tsai. Coherent control of macroscopic quantum states in a single-Cooper-pair box. In: *Nature* 398.6730 (1999), pp. 786–788.

[22] Isaac L. Chuang, Neil Gershenfeld, and Mark Kubinec. Experimental implementation of fast quantum searching. In: *Physical Review Letters* 80.15 (1998), p. 3408. URL: https://doi.org/10.1103/PhysRevLett.80.3408.

[23] Chris Monroe et al. Demonstration of a fundamental quantum logic gate. In: *Physical Review Letters* 75.25 (1995), p. 4714. URL: https://doi.org/10.1103/PhysRevLett.75.4714.

[24] Iulia Buluta, Sahel Ashhab, and Franco Nori. Natural and artificial atoms for quantum computation. In: *Reports on Progress in Physics* 74.10 (2011), p. 104401. URL: doi.org/10.1088/0034-4885/74/10/104401.

[25] Dario Gil and William M.J. Green. 1.4 the future of computing: Bits+ neurons+ qubits. In: *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE. 2020, pp. 30–39. URL: https://doi.org/10.1109/ISSCC19947.2020.9062918.

[26] Antonio D. Córcoles et al. Challenges and opportunities of near-term quantum computing systems. In: *Proceedings of the IEEE* 108.8 (2019), pp. 1338–1352. URL: https://doi.org/10.1109/JPROC.2019.2954005.

[27] Aaron Somoroff et al. Millisecond coherence in a superconducting qubit. In: *Physical Review Letters* 130.26 (2023), p. 267001. URL: https://doi.org/10.1103/PhysRevLett.130.267001.

[28] Yuan Xu et al. High-fidelity, high-scalability two-qubit gate scheme for superconducting qubits. In: *Physical Review Letters* 125.24 (2020), p. 240503. URL: https://doi.org/10.1103/PhysRevLett.125.240503.

[29] Ilya N. Moskalenko et al. High fidelity two-qubit gates on fluxoniums using a tunable coupler. In: *npj Quantum Information* 8.1 (2022), p. 130. URL: https://doi.org/10.1038/s41534-022-00644-x.

[30] Laird Egan et al. Fault-tolerant control of an error-corrected qubit. In: *Nature* 598.7880 (2021), pp. 281–286. URL: https://doi.org/10.1038/s41586-021-03928-y.

[31] Ronald L. Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. In: *Communications of the ACM* 21.2 (1978), pp. 120–126. URL: https://doi.org/10.1145/359340.359342.

[32] Daniel J Bernstein and Tanja Lange. Post-quantum cryptography. In: *Nature* 549.7671 (2017), pp. 188–194. URL: https://doi.org/10.1038/nature23461.

[33] Yudong Cao, Jhonathan Romero, and Alán Aspuru-Guzik. Potential of quantum computing for drug discovery. In: *IBM Journal of Research and Development* 62.6 (2018), pp. 6–1. URL: https://doi.org/10.1147/JRD.2018.2888987.

[34] Nick S. Blunt et al. Perspective on the current state-of-the-art of quantum computing for drug discovery applications. In: *Journal of Chemical Theory and Computation* 18.12 (2022), pp. 7001–7023. URL: https://doi.org/10.1021/acs.jctc.2c00574.

[35] Bela Bauer et al. Quantum algorithms for quantum chemistry and quantum materials science. In: *Chemical Reviews* 120.22 (2020), pp. 12685–12717. URL: https://doi.org/10.1021/acs.chemrev.9b00829.

[36] Nathalie P. De Leon et al. Materials challenges and opportunities for quantum computing hardware. In: *Science* 372.6539 (2021), eabb2823. URL: https://doi.org/10.1126/science.abb2823.

[37] Román Orús, Samuel Mugel, and Enrique Lizaso. Quantum computing for finance: Overview and prospects. In: *Reviews in Physics* 4 (2019), p. 100028. URL: https://doi.org/10.1016/j.revip.2019.100028.

[38] Daniel J. Egger et al. Quantum computing for finance: State-of-the-art and future prospects. In: *IEEE Transactions on Quantum Engineering* 1 (2020), pp. 1–24. URL: https://doi.org/10.1109/TQE.2020.3030314.

[39] Patrick Rebentrost and Seth Lloyd. Quantum computational finance: quantum algorithm for portfolio optimization. In: *arXiv preprint arXiv:1811.03975* (2018). URL: https://arxiv.org/abs/1811.03975.

[40] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. In: *Contemporary Physics* 56.2 (2015), pp. 172–185. URL: https://doi.org/10.1080/00107514.2014.964942.

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

[42] Yao Zhang and Qiang Ni. Recent advances in quantum machine learning. In: *Quantum Engineering* 2.1 (2020), e34. URL: https://doi.org/10.1002/que2.34.

[43] Robert Raussendorf and Hans J. Briegel. A one-way quantum computer. In: *Physical Review Letters* 86.22 (2001), p. 5188. URL: https://doi.org/10.1103/PhysRevLett.86.5188.

[44] Alexei Y. Kitaev. Fault-tolerant quantum computation by anyons. In: *Annals of physics* 303.1 (2003), pp. 2–30. URL: https://doi.org/10.1016/S0003-4916(02)00018-0.

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

[46] Xiu Gu et al. Fast multiqubit gates through simultaneous two-qubit gates. In: *PRX Quantum* 2.4 (2021), p. 040348. URL: https://doi.org/10.1103/PRXQuantum.2.040348.

[47] Tanay Roy et al. Programmable superconducting processor with native three-qubit gates. In: *Physical Review Applied* 14.1 (2020), p. 014072. URL: https://doi.org/10.1103/PhysRevApplied.14.014072.

[48] Seth Lloyd. Almost any quantum logic gate is universal. In: *Physical Review Letters* 75.2 (1995), p. 346. URL: https://doi.org/10.1103/PhysRevLett.75.346.

[49] Christopher M. Dawson and Michael A. Nielsen. The Solovay-Kitaev algorithm. In: *Quantum Information and Computation* 6.81 (2006). arXiv: quant-ph/0505030. URL: http://arxiv.org/abs/quant-ph/0505030.

[50] Paul V. Klimov et al. Fluctuations of energy-relaxation times in superconducting qubits. In: *Physical Review Letters* 121.9 (2018), p. 090502. URL: https://doi.org/10.1103/PhysRevLett.121.090502.

[51] Sebastian De Graaf et al. Direct identification of dilute surface spins on Al 2 O 3: Origin of flux noise in quantum circuits. In: *Physical Review Letters* 118.5 (2017), p. 057703. URL: https://doi.org/10.1103/PhysRevLett.118.057703.

[52] Jürgen Lisenfeld et al. Electric field spectroscopy of material defects in transmon qubits. In: *npj Quantum Information* 5.1 (2019), p. 105. URL: https://doi.org/10.1038/s41534-019-0224-1.

[53] Daniel Greenbaum and Zachary Dutton. Modeling coherent errors in quantum error correction. In: *Quantum Science and Technology* 3.1 (2017), p. 015007. URL: doi.org/10.1088/2058-9565/aa9a06.

[54] Sergey Bravyi et al. Correcting coherent errors with surface codes. In: *npj Quantum Information* 4.1 (2018), p. 55. URL: https://doi.org/10.1038/s41534-018-0106-y.

[55] Sergey Bravyi et al. Mitigating measurement errors in multi-qubit experiments. In: *arXiv:2006.14044* (2020). URL: https://arxiv.org/abs/2006.14044.

[56] Kristan Temme, Sergey Bravyi, and Jay M. Gambetta. Error mitigation for short-depth quantum circuits. In: *Physical Review Letters* 119.18 (2017), p. 180509. URL: https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.119.180509.

[57] Suguru Endo, Simon C. Benjamin, and Ying Li. Practical quantum error mitigation for near-future applications. In: *Physical Review X* 8.3 (2018), p. 031027. URL: https://doi.org/10.1103/PhysRevX.8.031027.

[58] Hirsh Kamakari et al. Digital quantum simulation of open quantum systems using quantum imaginary–time evolution. In: *PRX Quantum* 3.1 (2022), p. 010320. URL: https://doi.org/10.1103/PRXQuantum.3.010320.

[59] Mario Motta et al. Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution. In: *Nature Physics* 16.2 (Feb. 2020), pp. 205–210. ISSN: 1745-2473, 1745-2481. DOI: 10.1038/s41567-019-0704-4.

[60] Jin Ming Koh et al. Measurement-induced entanglement phase transition on a superconducting quantum processor with mid-circuit readout. In: *Nature Physics* 19.19 (2023), pp. 1314–1319. URL: https://doi.org/10.1038/s41567-023-02076-6.

[61] Yaodong Li et al. Cross entropy benchmark for measurement-induced phase transitions. In: *Physical Review Letters* 130.22 (2023), p. 220404. URL: https://doi.org/10.1103/PhysRevLett.130.220404.

[62] Mithuna Yoganathan, Richard Jozsa, and Sergii Strelchuk. Quantum advantage of unitary Clifford circuits with magic state inputs. In: *Proceedings of the Royal Society A* 475.2225 (2019), p. 20180427. URL: https://doi.org/10.1098/rspa.2018.0427.

[63] Heinz-Peter Breuer and Francesco Petruccione. The theory of open quantum systems. Oxford University Press, 2002. ISBN: 978-0-19-852063-4. DOI: 10.1093/acprof:oso/9780199213900.001.0001.

[64] Marco Cerezo et al. Variational Quantum Algorithms. 2020. arXiv: 2012.09265 [quant-ph].

[65] Iulia M. Georgescu, Sahel Ashhab, and Franco Nori. Quantum simulation. In: *Reviews of Modern Physics* 86.1 (2014), p. 153. DOI: 10.1103/RevModPhys.86.153.

[66] Kishor Bharti et al. Noisy intermediate-scale quantum (NISQ) algorithms. In: *arXiv preprint arXiv:2101.08448* (2021). URL: https://arxiv.org/abs/2101.08448v1.

[67] Benedikt Fauseweh and Jian-Xin Zhu. Digital quantum simulation of non-equilibrium quantum many-body systems. In: *arXiv:2009.07375* (2020). URL: https://arxiv.org/abs/2009.07375v2.

[68] Adam Smith et al. Simulating quantum many-body dynamics on a current digital quantum computer. In: *npj Quantum Information* 5.1 (2019), pp. 1–13. DOI: https://doi.org/10.1038/s41534-019-0217-0.

[69] Allesandro Chiesa et al. Quantum hardware simulating four-dimensional inelastic neutron scattering. In: *Nature Physics* 15.5 (2019), pp. 455–459. DOI: https://doi.org/10.1038/s41567-019-0437-4.

[70] Henry Lamm and Scott Lawrence. Simulation of nonequilibrium dynamics on a quantum computer. In: *Physical Review Letters* 121.17 (2018), p. 170501. DOI: https://doi.org/10.1103/PhysRevLett.121.170501.

[71] Suguru Endo et al. Variational quantum simulation of general processes. In: *Physical Review Letters* 125.1 (June 2020), p. 010501. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.125.010501.

[72] Cristina Cirstoiu et al. Variational fast forwarding for quantum simulation beyond the coherence time. In: *npj Quantum Information* 6.1 (2020), pp. 1–10. DOI: https://doi.org/10.1038/s41534-020-00302-0.

[73] Joe Gibbs et al. Long-time simulations with high fidelity on quantum hardware. In: *arXiv preprint arXiv:2102.04313* (2021). URL: https://arxiv.org/abs/2102.04313v1.

[74] Rami Barends et al. Digital quantum simulation of fermionic models with a superconducting circuit. In: *Nature Communications* 6.1 (2015), pp. 1–7. DOI: https://doi.org/10.1038/ncomms8654.

[75] Frank Arute et al. Observation of separated dynamics of charge and spin in the Fermi-Hubbard model. In: *arXiv:2010.07965* (2020). URL: https://arxiv.org/abs/2010.07965.

[76] Alexandru Macridin et al. Digital quantum computation of fermion-boson interacting systems. In: *Physical Review A* 98 (4 Oct. 2018), p. 042312. DOI: 10.1103/PhysRevA.98.042312. URL: https://link.aps.org/doi/10.1103/PhysRevA.98.042312.

[77] Stephen P. Jordan, Keith S.M. Lee, and John Preskill. Quantum algorithms for quantum field theories. In: *Science* 336.6085 (2012), pp. 1130–1133. DOI: 10.1126/science.1217069.

[78] Dmitri E. Kharzeev and Yuta Kikuchi. Real-time chiral dynamics from a digital quantum simulation. In: *Physical Review Research* 2.2 (2020), p. 023342. DOI: 10.1103/PhysRevResearch.2.023342.

[79] Michael Kreshchuk et al. Quantum simulation of quantum field theory in the light-front formulation. In: *arXiv:2002.04016* (2020). URL: https://arxiv.org/abs/2002.04016v2.

[80] Daniel A. Lidar. Lecture notes on the theory of open quantum systems. In: *arXiv:1902.00967* (2019). URL: https://arxiv.org/abs/1902.00967v2.

[81] Chun-hsin Tseng et al. Quantum simulation with natural decoherence. In: *Physical Review A* 62.3 (Aug. 2000), p. 032309. ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.62.032309.

[82] Hefeng Wang, S. Ashhab, and Franco Nori. Quantum algorithm for simulating the dynamics of an open quantum system. In: *Physical Review A* 83.6 (June 2011), p. 062317. ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.83.062317.

[83] Hong-Yi Su and Ying Li. Quantum algorithm for the simulation of open-system dynamics and thermalization. In: *Physical Review A* 101 (1 Jan. 2020), p. 012328. DOI: 10.1103/PhysRevA.101.012328. URL: https://link.aps.org/doi/10.1103/PhysRevA.101.012328.

[84] Marco Cattaneo et al. Collision models can efficiently simulate any multipartite Markovian quantum dynamics. In: *Physical Review Letters* 126 (13 Apr. 2021), p. 130403. DOI: 10.1103/PhysRevLett.126.130403. URL: https://link.aps.org/doi/10.1103/PhysRevLett.126.130403.

[85] Dave Bacon et al. Universal simulation of Markovian quantum dynamics. In: *Physical Review A* 64.6 (Nov. 2001), p. 062302. ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.64.062302.

[86] Ryan Sweke et al. Universal simulation of Markovian open quantum systems. In: *Physical Review A* 91.6 (June 2015), p. 062308. ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.91.062308.

[87] Martin Kliesch et al. Dissipative quantum Church-Turing theorem. In: *Physical Review Letters* 107 (Sept. 2011), p. 120501. DOI: 10.1103/PhysRevLett.107.120501.

[88] Shi-Jie Wei, Dong Ruan, and Gui-Lu Long. Duality quantum algorithm efficiently simulates open quantum systems. In: *Scientific Reports* 6.1 (Aug. 2016), p. 30727. ISSN: 2045-2322. DOI: 10.1038/srep30727.

[89] Zixuan Hu, Rongxin Xia, and Sabre Kais. A quantum algorithm for evolving open quantum dynamics on quantum computing devices. In: *Scientific Reports* 10.1 (Dec. 2020), p. 3301. ISSN: 2045-2322. DOI: 10.1038/s41598-020-60321-x.

[90] Jay Hubisz, Bharath Sambasivam, and Judah Unmuth-Yockey. Quantum algorithms for open lattice field theory. In: *arXiv:2012.05257* (2020). URL: https://arxiv.org/abs/2012.05257v1.

[91] Zixuan Hu et al. A general quantum algorithm for open quantum dynamics demonstrated with the Fenna-Matthews-Olson complex. In: *arXiv preprint arXiv:2101.05287* (2021). URL: https://arxiv.org/abs/2101.05287v1.

[92] Kade Head-Marsden et al. Capturing non-Markovian dynamics on near-term quantum computers. In: *Phys. Rev. Research* 3 (1 Feb. 2021), p. 013182. DOI: 10.1103/PhysRevResearch.3.013182. URL: https://link.aps.org/doi/10.1103/PhysRevResearch.3.013182.

[93] Lorenzo Del Re et al. Driven-dissipative quantum mechanics on a lattice: Simulating a fermionic reservoir on a quantum computer. In: *Phys. Rev. B* 102 (12 Sept. 2020), p. 125112. DOI: 10.1103/PhysRevB.102.125112. URL: https://link.aps.org/doi/10.1103/PhysRevB.102.125112.

[94] Tobias Haug and Kishor Bharti. Generalized quantum assisted simulator. In: *arXiv preprint arXiv:2011.14737* (2020). URL: https://arxiv.org/abs/2011.14737v1.

[95] Jiaxiu Han et al. Experimental simulation of open quantum system dynamics via Trotterization. In: *Phys. Rev. Lett.* 127 (2 July 2021), p. 020504. DOI: 10.1103/PhysRevLett.127.020504. URL: https://link.aps.org/doi/10.1103/PhysRevLett.127.020504.

[96] Erika Andersson, James D. Cresser, and Michael J. W. Hall. Finding the Kraus decomposition from a master equation and vice versa. In: *Journal of Modern Optics* 54.12 (Aug. 2007), pp. 1695–1716. ISSN: 0950-0340, 1362-3044. DOI: 10.1080/09500340701352581.

[97] Nobuyuki Yoshioka et al. Variational quantum algorithm for nonequilibrium steady states. In: *Physical Review Research* 2.4 (Nov. 2020), p. 043289. ISSN: 2643-1564. DOI: 10.1103/PhysRevResearch.2.043289.

[98] Lennart Bittel and Martin Kliesch. Training variational quantum algorithms is NP-Hard. In: *Physical Review Letters* 127 (12 Sept. 2021), p. 120502. DOI: 10.1103/PhysRevLett.127.120502. URL: https://link.aps.org/doi/10.1103/PhysRevLett.127.120502.

[99] Sam McArdle et al. Variational ansatz-based quantum simulation of imaginary time evolution. In: *npj Quantum Information* 5.1 (Dec. 2019), p. 75. ISSN: 2056-6387. DOI: 10.1038/s41534-019-0187-2.

[100] Shi-Ning Sun et al. Quantum computation of finite-temperature static and dynamical properties of spin systems using quantum imaginary time evolution. In: *PRX Quantum* 2 (1 Feb. 2021), p. 010317. DOI: 10.1103/PRXQuantum.2.010317. URL: https://link.aps.org/doi/10.1103/PRXQuantum.2.010317.

[101] Kubra Yeter Aydeniz, George Siopsis, and Raphael C. Pooser. Scattering in the Ising model with the quantum Lanczos algorithm. In: *New Journal of Physics* 23 (Feb. 2021), p. 043033. ISSN: 1367-2630. URL: https://doi.org/10.1088/1367-2630/abe63d.

[102] Niladri Gomes et al. Efficient step-merged quantum imaginary time evolution algorithm for quantum chemistry. In: *Journal of Chemical Theory and Computation* 16.10 (Oct. 2020), pp. 6256–6266. ISSN: 1549-9618, 1549-9626. DOI: 10.1021/acs.jctc.0c00666.

[103] Kubra Yeter-Aydeniz, Raphael C. Pooser, and George Siopsis. Practical quantum computation of chemical and nuclear energy levels using quantum imaginary time evolution and Lanczos algorithms. In: *npj Quantum Information* 6.1 (Dec. 2020), p. 63. ISSN: 2056-6387. DOI: 10.1038/s41534-020-00290-1.

[104] Goran Lindblad. On the generators of quantum dynamical semigroups. In: *Communications in Mathematical Physics* 48.2 (1976), pp. 119–130. DOI: https://doi.org/10.1007/BF01608499.

[105] Timothy F. Havel. Robust procedures for converting among Lindblad, Kraus and matrix representations of quantum dynamical semigroups. In: *Journal of Mathematical Physics* 44.2 (Feb. 2003), p. 534. ISSN: 00222488. DOI: 10.1063/1.1518555. URL: https://doi.org/10.1063/1.1518555.

[106] Rolando Somma et al. Simulating physical phenomena by quantum networks. In: *Phys. Rev. A* 65 (4 Apr. 2002), p. 042323. DOI: 10.1103/PhysRevA.65.042323. URL: https://link.aps.org/doi/10.1103/PhysRevA.65.042323.

[107] Suguru Endo et al. Hybrid quantum-classical algorithms and quantum error mitigation. In: *Journal of the Physical Society of Japan* 90.3 (2021), p. 032001. URL: https://journals.jps.jp/doi/abs/10.7566/JPSJ.90.032001.

[108] Robert M. Parrish et al. Quantum computation of electronic transitions using a variational quantum eigensolver. In: *Physical Review Letters* 122.23 (2019), p. 230401. URL: https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.122.230401.

[109] Sergey Bravyi, Graeme Smith, and John A. Smolin. Trading classical and quantum computational resources. In: *Phys. Rev. X* 6 (2 June 2016), p. 021043. DOI: 10.1103/PhysRevX.6.021043. URL: https://link.aps.org/doi/10.1103/PhysRevX.6.021043.

[110] Tianyi Peng et al. Simulating large quantum circuits on a small quantum computer. In: *Phys. Rev. Lett.* 125 (15 Oct. 2020), p. 150504. DOI: 10.1103/PhysRevLett.125.150504. URL: https://link.aps.org/doi/10.1103/PhysRevLett.125.150504.

[111] Tanvi Gujarati et al. Reducing circuit size in the variational quantum eigensolver–Part 1: Theory. In: *Bulletin of the American Physical Society* (2021). URL: https://meetings.aps.org/Meeting/MAR21/Session/M32.7.

[112] Andrew Eddins et al. Reducing circuit size in the variational quantum eigensolver–Part 2: Experiment. In: *Bulletin of the American Physical Society* (2021). URL: https://meetings.aps.org/Meeting/MAR21/Session/M32.8.

[113] Andrew Eddins et al. Doubling the size of quantum simulators by entanglement forging. In: *arXiv:2104.10220* (2021). URL: https://arxiv.org/abs/2104.10220.

[114] Masuo Suzuki. Generalized Trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems. In: *Communications in Mathematical Physics* 51.2 (1976), pp. 183–190. URL: https://doi.org/10.1007/BF01609348.

[115] Michael J. Kastoryano and Jens Eisert. Rapid mixing implies exponential decay of correlations. In: *Journal of Mathematical Physics* 54.10 (2013), p. 102201. URL: https://doi.org/10.1063/1.4822481.

[116] Gadi Aleksandrowicz et al. Qiskit: An open-source framework for quantum computing. https://doi.org/10.5281/zenodo.2562110. Accessed: 2021-03-16. 2019.

[117] Abhinav Kandala et al. Error mitigation extends the computational reach of a noisy quantum processor. In: *Nature* 567.7749 (2019), pp. 491–495. URL: https://www.nature.com/articles/s41586-019-1040-7?platform=hootsuite.

[118] *ibmq_mumbai* v1.4.11, *ibmq_guadalupe* v1.2.14, and *ibmq_casablanca* v1.2.14. IBM Quantum Team, Retrieved from https://quantum-computing.ibm.com. Accessed: 2021-03-16. 2020.

[119] Robert Johansson, Paul .D. Nation, and Franco Nori. QuTiP 2: A Python framework for the dynamics of open quantum systems. In: *Computer Physics Communications* 184.4 (2013), pp. 1234–1240. ISSN: 0010-4655. DOI: https://doi.org/10.1016/j.cpc.2012.11.019. URL: https://www.sciencedirect.com/science/article/pii/S0010465512003955.

[120] Robert Johansson, Paul D. Nation, and Franco Nori. QuTiP: An open-source Python framework for the dynamics of open quantum systems. In: *Computer Physics Communications* 183.8 (2012), pp. 1760–1772. DOI: http://dx.doi.org/10.1016/j.cpc.2012.11.019.

[121] Rodney Loudon. The quantum theory of light. Oxford University Press, 2000.

[122] Yunchao Liu et al. Benchmarking near-term quantum computers via random circuit sampling. In: *arXiv preprint arXiv:2105.05232* (2021). URL: https://arxiv.org/abs/2105.05232.

[123] Frank Arute et al. Quantum supremacy using a programmable superconducting processor. In: *Nature* 574.7779 (2019), pp. 505–510. URL: https://doi.org/10.1038/s41586-019-1666-5.

[124] Brian Skinner, Jonathan Ruhman, and Adam Nahum. Measurement-induced phase transitions in the dynamics of entanglement. In: *Phys. Rev. X* 9.3, 031009 (July 2019), p. 031009. DOI: 10.1103/PhysRevX.9.031009.

[125] Amos Chan et al. Unitary-projective entanglement dynamics. In: *Physical Review B* 99.22 (2019), p. 224307. URL: https://doi.org/10.1103/PhysRevB.99.224307.

[126] Yaodong Li, Xiao Chen, and Matthew PA Fisher. Quantum Zeno effect and the many-body entanglement transition. In: *Physical Review B* 98.20 (2018), p. 205136. URL: https://doi.org/10.1103/PhysRevB.98.205136.

[127] Pasquale Calabrese and John Cardy. Evolution of entanglement entropy in one-dimensional systems. In: *Journal of Statistical Mechanics: Theory and Experiment* 2005.04 (2005), P04010. URL: http://dx.doi.org/10.1088/1742-5468/2005/04/P04010.

[128] Márk Mezei and Douglas Stanford. On entanglement spreading in chaotic systems. In: *Journal of High Energy Physics* 2017, 65 (May 2017), p. 65. DOI: 10.1007/JHEP05(2017)065. arXiv: 1608.05101 [hep-th].

[129] Adam Nahum, Sagar Vijay, and Jeongwan Haah. Operator Spreading in Random Unitary Circuits. In: *Phys. Rev. X* 8, 021014 (Apr. 2018), p. 021014. DOI: 10.1103/PhysRevX.8.021014.

[130] Curt W. von Keyserlingk et al. Operator Hydrodynamics, OTOCs, and Entanglement Growth in Systems without Conservation Laws. In: *Phys. Rev. X* 8, 021013 (Apr. 2018), p. 021013. DOI: 10.1103/PhysRevX.8.021013. arXiv: 1705.08910 [cond-mat.str-el].

[131] Xiangyu Cao, Antoine Tilloy, and Andrea De Luca. Entanglement in a fermion chain under continuous monitoring. In: *SciPost Physics* 7.2, 024 (Aug. 2019), p. 024. DOI: 10.21468/SciPostPhys.7.2.024.

[132] Yaodong Li, Xiao Chen, and Matthew P.A. Fisher. Measurement-driven entanglement transition in hybrid quantum circuits. In: *Physical Review B* 100.13 (2019), p. 134306. URL: https://doi.org/10.1103/PhysRevB.100.134306.

[133] Adam Nahum and Brian Skinner. Entanglement and dynamics of diffusion-annihilation processes with Majorana defects. In: *Phys. Rev. Research* 2.2, 023288 (June 2020), p. 023288. DOI: 10.1103/PhysRevResearch.2.023288.

[134] Javier Lopez-Piqueres, Brayden Ware, and Romain Vasseur. Mean-field entanglement transitions in random tree tensor networks. In: *Phys. Rev. B* 102 (6 Aug. 2020), p. 064202. DOI: 10.1103/PhysRevB.102.064202. URL: https://link.aps.org/doi/10.1103/PhysRevB.102.064202.

[135] Ali Lavasani, Yahya Alavirad, and Maissam Barkeshli. Measurement-induced topological entanglement transitions in symmetric random quantum circuits. In: *Nature Physics* 17.3 (Jan. 2021), pp. 342–347. DOI: 10.1038/s41567-020-01112-z.

[136] Shengqi Sang and Timothy H. Hsieh. Measurement-protected quantum phases. In: *Phys. Rev. Research* 3.2, 023200 (June 2021), p. 023200. DOI: 10.1103/PhysRevResearch.3.023200.

[137] Matteo Ippoliti et al. Entanglement phase transitions in measurement-only dynamics. In: *Phys. Rev. X* 11.1, 011030 (Jan. 2021), p. 011030. DOI: 10.1103/PhysRevX.11.011030.

[138] Xiao Chen et al. Emergent conformal symmetry in nonunitary random dynamics of free fermions. In: *Phys. Rev. Research* 2.3, 033017 (July 2020), p. 033017. DOI: 10.1103/PhysRevResearch.2.033017.

[139] Yohei Fuji and Yuto Ashida. Measurement-induced quantum criticality under continuous monitoring. In: *Phys. Rev. B* 102 (5 Aug. 2020), p. 054302. DOI: 10.1103/PhysRevB.102.054302. URL: https://link.aps.org/doi/10.1103/PhysRevB.102.054302.

[140] Ori Alberton, Michael Buchhold, and Sebastian Diehl. Entanglement transition in a monitored free-fermion chain: From extended criticality to area law. In: *Physical Review Letters* 126.17 (2021), p. 170602. URL: https://doi.org/10.1103/PhysRevLett.126.170602.

[141] Sagar Vijay. Measurement-Driven Phase Transition within a Volume-Law Entangled Phase. 2020. arXiv: 2005.03052 [quant-ph].

[142] Adam Nahum et al. Measurement and entanglement phase transitions in all-To-all quantum circuits, on quantum trees, and in Landau-Ginsburg theory. In: *PRX Quantum* 2 (1 Mar. 2021), p. 010352. DOI: 10.1103/PRXQuantum.2.010352. URL: https://link.aps.org/doi/10.1103/PRXQuantum.2.010352.

[143] Yimu Bao, Soonwon Choi, and Ehud Altman. Symmetry enriched phases of quantum circuits. In: *Annals of Physics* 435 (Dec. 2021), p. 168618. ISSN: 0003-4916. DOI: 10.1016/j.aop.2021.168618. URL: http://dx.doi.org/10.1016/j.aop.2021.168618.

[144] Utkarsh Agrawal et al. Entanglement and charge-sharpening transitions in U(1) symmetric monitored quantum circuits. In: *Phys. Rev. X* 12 (4 Oct. 2022), p. 041002. DOI: 10.1103/PhysRevX.12.041002. URL: https://link.aps.org/doi/10.1103/PhysRevX.12.041002.

[145] Fergus Barratt et al. Field theory of charge sharpening in symmetric monitored quantum circuits. In: *Phys. Rev. Lett.* 129 (12 Sept. 2022), p. 120604. DOI: 10.1103/PhysRevLett.129.120604. URL: https://link.aps.org/doi/10.1103/PhysRevLett.129.120604.

[146] Matthew P.A. Fisher et al. Random quantum circuits. In: *Annual Review of Condensed Matter Physics* 14.1 (Mar. 2023), pp. 335–379. ISSN: 1947-5462. DOI: 10.1146/annurev-conmatphys-031720-030658. URL: http://dx.doi.org/10.1146/annurev-conmatphys-031720-030658.

[147] Soonwon Choi et al. Quantum error correction in scrambling dynamics and measurement-induced phase transition. In: *Physical Review Letters* 125.3 (2020), p. 030505. URL: https://doi.org/10.1103/PhysRevLett.125.030505.

[148] Yimu Bao, Soonwon Choi, and Ehud Altman. Theory of the phase transition in random unitary circuits with measurements. In: *Physical Review B* 101.10 (2020), p. 104301. URL: https://doi.org/10.1103/PhysRevB.101.104301.

[149] Michael J. Gullans and David A. Huse. Localization as an entanglement phase transition in boundary-driven Anderson models. In: *Physical Review Letters* 123.11 (2019), p. 110601. URL: https://doi.org/10.1103/PhysRevLett.123.110601.

[150] Michael J. Gullans et al. Quantum coding with low-depth random circuits. In: *Physical Review X* 11.3 (2021), p. 031066. URL: https://doi.org/10.1103/PhysRevX.11.031066.

[151] John Napp et al. Efficient classical simulation of random shallow 2D quantum circuits. In: *arXiv e-prints*, arXiv:2001.00021 (Dec. 2019), arXiv:2001.00021. arXiv: 2001.00021 [quant-ph].

[152] Matteo Ippoliti and Vedika Khemani. Postselection-free entanglement dynamics via spacetime duality. In: *Physical Review Letters* 126.6 (2021), p. 060501. URL: https://doi.org/10.1103/PhysRevLett.126.060501.

[153] Tsung-Cheng Lu and Tarun Grover. Spacetime duality between localization transitions and measurement-induced transitions. In: *PRX Quantum* 2.4 (2021), p. 040319. URL: https://doi.org/10.1103/PRXQuantum.2.040319.

[154] Jesse C. Hoke et al. Measurement-induced entanglement and teleportation on a noisy quantum processor. In: *Nature* 622.7983 (2023), pp. 481–486. URL: https://doi.org/10.1038/s41586-023-06505-7.

[155] Michael J. Gullans and David A. Huse. Scalable probes of measurement-induced criticality. In: *Physical Review Letters* 125.7 (2020), p. 070606. URL: https://doi.org/10.1103/PhysRevLett.125.070606.

[156] Crystal Noel et al. Measurement-induced quantum phases realized in a trapped-ion quantum computer. In: *Nature Physics* 18.7 (June 2022), pp. 760–764. ISSN: 1745-2481. DOI: 10.1038/s41567-022-01619-7. URL: http://dx.doi.org/10.1038/s41567-022-01619-7.

[157] Sergio Boixo et al. Characterizing quantum supremacy in near-term devices. In: *Nature Physics* 14.6 (Apr. 2018), pp. 595–600. ISSN: 1745-2481. DOI: 10.1038/s41567-018-0124-x. URL: http://dx.doi.org/10.1038/s41567-018-0124-x.

[158] Brayden Ware et al. A sharp phase transition in linear cross-entropy benchmarking. 2023. arXiv: 2305.04954 [quant-ph].

[159] Alexis Morvan et al. Phase transition in Random Circuit Sampling. 2023. arXiv: 2304.11119 [quant-ph].

[160] Samuel J. Garratt, Zack Weinstein, and Ehud Altman. Measurements conspire nonlocally to restructure critical quantum states. In: *Phys. Rev. X* 13 (2 May 2023), p. 021026. DOI: 10.1103/PhysRevX.13.021026. URL: https://link.aps.org/doi/10.1103/PhysRevX.13.021026.

[161] Xiaozhou Feng, Brian Skinner, and Adam Nahum. Measurement-induced phase transitions on dynamical quantum trees. 2022. arXiv: 2210.07264 [cond-mat.stat-mech].

[162] Richard Jozsa and Marrten Van Den Nest. Classical simulation complexity of extended clifford circuits. In: *Quantum Information and Computation* 14.7 & 8 (May 2014), pp. 633–648. ISSN: 1533-7146.

[163] Dax Enshan Koh. Further extensions of Clifford circuits and their classical simulation complexities. In: *arXiv preprint arXiv:1512.07892* (2015). URL: https://arxiv.org/abs/1512.07892.

[164] Adam Bouland, Joseph F. Fitzsimons, and Dax Enshan Koh. Complexity Classification of Conjugated Clifford Circuits. 2017. arXiv: 1709.01805 [quant-ph].

[165] Sergey Bravyi and David Gosset. Improved classical simulation of quantum circuits dominated by Clifford gates. In: *Physical Review Letters* 116.25 (2016), p. 250501. URL: https://doi.org/10.1103/PhysRevLett.116.250501.

[166] Eugene Stanley. Scaling, universality, and renormalization: Three pillars of modern critical phenomena. In: *Reviews of modern physics* 71.2 (1999), S358. URL: https://doi.org/10.1103/RevModPhys.71.S358.

[167] Somendra M. Bhattacharjee and Flavio Seno. A measure of data collapse for scaling. In: *Journal of Physics A: Mathematical and General* 34.33 (2001), p. 6375. URL: https://dx.doi.org/10.1088/0305-4470/34/33/302.

[168] Adam Nahum et al. Measurement and entanglement phase transitions in all-to-all quantum circuits, on quantum trees, and in Landau-Ginsburg theory. In: *PRX Quantum* 2.1 (2021), p. 010352. URL: https://doi.org/10.1103/PRXQuantum.2.010352.

[169] Pauli Virtanen et al. SciPy 1.0: Fundamental algorithms for scientific computing in python. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[170] Lorenza Viola and Seth Lloyd. Dynamical suppression of decoherence in two-state quantum systems. In: *Physical Review A* 58.4 (1998), p. 2733. URL: https://doi.org/10.1103/PhysRevA.58.2733.

[171] Abraham G. Kofman and Gershon Kurizki. Universal dynamical control of quantum mechanical decay: modulation of the coupling to the continuum. In: *Physical Review Letters* 87.27 (2001), p. 270405. URL: https://doi.org/10.1103/PhysRevLett.87.270405.

[172] Michael J. Biercuk et al. Optimized dynamical decoupling in a model quantum memory. In: *Nature* 458.7241 (2009), pp. 996–1000. URL: https://doi.org/10.1038/nature07951.

[173] Brian Rost et al. Simulation of thermal relaxation in spin chemistry systems on a quantum computer using inherent qubit decoherence. In: *arXiv:2001.00794* (2020). URL: https://arxiv.org/abs/2001.00794.

[174] Siyuan Niu and Aida Todri-Sanial. Effects of dynamical decoupling and pulse-level optimizations on ibm quantum computers. In: *IEEE Transactions on Quantum Engineering* 3 (2022), pp. 1–10. URL: https://doi.org/10.1109/TQE.2022.3203153.

[175] Siyuan Niu and Aida Todri-Sanial. Analyzing strategies for dynamical decoupling insertion on IBM quantum computer. In: *arXiv:2204.14251* (2022). URL: https://arxiv.org/abs/2204.14251.

[176] Nic Ezzell et al. Dynamical decoupling for superconducting qubits: a performance survey. In: *Physical Review Applied* 20.6 (2023), p. 064027. URL: https://doi.org/10.1103/PhysRevApplied.20.064027.

[177] Sergey Bravyi et al. Mitigating measurement errors in multiqubit experiments. In: *Physical Review A* 103.4 (2021), p. 042605. URL: https://doi.org/10.1103/PhysRevA.103.042605.

[178] Paul D. Nation et al. Scalable mitigation of measurement errors on quantum computers. In: *PRX Quantum* 2.4 (2021), p. 040326. URL: https://doi.org/10.1103/PRXQuantum.2.040326.

[179] Chao-Ming Jian et al. Measurement-induced criticality in random quantum circuits. In: *Physical Review B* 101.10 (2020), p. 104302. URL: https://doi.org/10.1103/PhysRevB.101.104302.

[180] Tianci Zhou and Adam Nahum. Emergent statistical mechanics of entanglement in random unitary circuits. In: *Physical Review B* 99.17 (2019), p. 174205. URL: https://doi.org/10.1103/PhysRevB.99.174205.

[181] Tianci Zhou and Adam Nahum. Entanglement membrane in chaotic many-body systems. In: *Phys. Rev. X* 10 (3 Sept. 2020), p. 031066. DOI: 10.1103/PhysRevX.10.031066. URL: https://link.aps.org/doi/10.1103/PhysRevX.10.031066.

[182] Shengqi Sang and Timothy H Hsieh. Measurement-protected quantum phases. In: *Physical Review Research* 3.2 (2021), p. 023200. URL: https://doi.org/10.1103/PhysRevResearch.3.023200.

[183] Ali Lavasani, Yahya Alavirad, and Maissam Barkeshli. Measurement-induced topological entanglement transitions in symmetric random quantum circuits. In: *Nature Physics* 17.3 (2021), pp. 342–347. URL: https://doi.org/10.1038/s41567-020-01112-z.

[184] Maria Tikhanovskaya et al. Universality of the cross entropy in $\mathbb{Z}_2$ symmetric monitored quantum circuits. 2023. arXiv: 2306.00058 [quant-ph].

[185] Izabella Lovas, Utkarsh Agrawal, and Sagar Vijay. Quantum Coding Transitions in the Presence of Boundary Dissipation. 2023. arXiv: 2304.02664 [quant-ph].

[186] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[187] Yaniv Taigman et al. DeepFace: Closing the gap to human-level performance in face verification. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2014.

[188] Alexey Dosovitskiy et al. An image is worth 16x16 words: Transformers for image recognition at scale. In: *arXiv preprint arXiv:2010.11929* (2020). URL: https://arxiv.org/abs/2010.11929.

[189] Tomas Mikolov et al. Efficient estimation of word representations in vector space. In: *arXiv preprint arXiv:1301.3781* (2013). URL: https://arxiv.org/abs/1301.3781.

[190] Ashish Vaswani et al. Attention is all you need. In: *Advances in Neural Information Processing Systems* 30 (2017).

[191] Tom Brown et al. Language models are few-shot learners. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1877–1901.

[192] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6840–6851.

[193] Ian Goodfellow et al. Generative adversarial nets. In: *Advances in Neural Information Processing Systems* 27 (2014).

[194] Sam McArdle et al. Quantum computational chemistry. In: *Reviews of Modern Physics* 92.1 (2020), p. 015003. URL: https://doi.org/10.1103/RevModPhys.92.015003.

[195] Mario Motta and Julia E. Rice. Emerging quantum computing algorithms for quantum chemistry. In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 12.3 (2022), e1580. URL: https://doi.org/10.1002/wcms.1580.

[196] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. In: *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*. 1984, p. 175. URL: https://doi.org/10.1016/j.tcs.2014.05.025.

[197] Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. A rigorous and robust quantum speed-up in supervised machine learning. In: *Nature Physics* 17.9 (2021), pp. 1013–1017. URL: https://doi.org/10.1038/s41567-021-01287-z.

[198] Ryan Sweke et al. On the quantum versus classical learnability of discrete distributions. In: *Quantum* 5 (2021), p. 417. URL: https://doi.org/10.22331/q-2021-03-23-417.

[199] Christian Szegedy et al. Intriguing properties of neural networks. In: *arXiv preprint arXiv:1312.6199* (2013). URL: https://arxiv.org/abs/1312.6199.

[200] Nicholas Carlini et al. Hidden voice commands. In: *25th USENIX Security Symposium (USENIX Security 16)*. 2016, pp. 513–530.

[201] Guoming Zhang et al. DolphinAttack: Inaudible voice commands. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. CCS '17. Dallas, Texas, USA: Association for Computing Machinery, 2017, pp. 103–117. ISBN: 9781450349468. DOI: 10.1145/3133956.3134052. URL: https://doi.org/10.1145/3133956.3134052.

[202] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In: *arXiv preprint arXiv:1611.01236* (2016). URL: https://arxiv.org/abs/1611.01236.

[203] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In: *arXiv preprint arXiv:1412.6572* (2014). URL: https://arxiv.org/abs/1412.6572.

[204] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In: *arXiv preprint arXiv:1503.02531* (2015). URL: https://arxiv.org/abs/1503.02531.

[205] Nicolas Papernot et al. Distillation as a defense to adversarial perturbations against deep neural networks. In: *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2016, pp. 582–597. URL: https://doi.org/10.1109/SP.2016.41.

[206] Anirban Chakraborty et al. Adversarial attacks and defences: A survey. In: *arXiv preprint arXiv:1810.00069* (2018). URL: https://arxiv.org/abs/1810.00069.

[207] Sébastien Bubeck et al. Adversarial examples from computational constraints. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 831–840. URL: https://proceedings.mlr.press/v97/bubeck19a.html.

[208] Akshay Degwekar, Preetum Nakkiran, and Vinod Vaikuntanathan. Computational limitations in robust classification and win-win results. In: *Proceedings of the Thirty-Second Conference on Learning Theory*. Ed. by Alina Beygelzimer and Daniel Hsu. Vol. 99. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 994–1028. URL: https://proceedings.mlr.press/v99/degwekar19a.html.

[209] Nathan Wiebe and Ram Shankar Siva Kumar. Hardening quantum machine learning against adversaries. In: *New Journal of Physics* 20.12 (2018), p. 123019. URL: doi.org/10.1088/1367-2630/aae71a.

[210] Chenyi Huang and Shibin Zhang. Enhancing adversarial robustness of quantum neural networks by adding noise layers. In: *New Journal of Physics* 25.8 (2023), p. 083019. URL: doi.org/10.1088/1367-2630/ace8b4.

[211] Yuxuan Du et al. Quantum noise protects quantum classifiers against adversaries. In: *Physical Review Research* 3.2 (2021), p. 023153. URL: https://doi.org/10.1103/PhysRevResearch.3.023153.

[212] Nana Liu and Peter Wittek. Vulnerability of quantum classification to adversarial perturbations. In: *Physical Review A* 101.6 (2020), p. 062331. URL: https://doi.org/10.1103/PhysRevA.101.062331.

[213] Haoran Liao et al. Robust in practice: Adversarial attacks on quantum machine learning. In: *Physical Review A* 103.4 (2021), p. 042427. URL: https://doi.org/10.1103/PhysRevA.103.042427.

[214] Maurice Weber et al. Optimal provable robustness of quantum classification via quantum hypothesis testing. In: *npj Quantum Information* 7.1 (2021), p. 76. URL: https://doi.org/10.1038/s41534-021-00410-5.

[215] Nader H. Bshouty and Jeffrey C. Jackson. Learning DNF over the uniform distribution using a quantum example oracle. In: *Proceedings of the Eighth Annual Conference on Computational Learning Theory*. 1995, pp. 118–127. URL: https://doi.org/10.1145/225298.225312.

[216] Khashayar Barooti, Grzegorz Głuch, and Ruediger Urbanke. Provable adversarial robustness in the quantum model. In: *arXiv preprint arXiv:2112.09625* (2021). URL: https://arxiv.org/abs/2112.09625.

[217] Maxwell T. West et al. Benchmarking adversarially robust quantum machine learning at scale. In: *Physical Review Research* 5.2 (2023), p. 023186. URL: https://doi.org/10.1103/PhysRevResearch.5.023186.

[218] Wenhui Ren et al. Experimental quantum adversarial learning with programmable superconducting qubits. In: *Nature Computational Science* 2.11 (2022), pp. 711–717. URL: https://doi.org/10.1038/s43588-022-00351-9.

[219] Dan Boneh. The decision Diffie-Hellman problem. In: *Algorithmic Number Theory*. Ed. by Joe P. Buhler. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 48–63. ISBN: 978-3-540-69113-6.

[220] Venkatesan Guruswami and Piotr Indyk. Expander-based constructions of efficiently decodable codes. In: *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*. IEEE. 2001, pp. 658–667. URL: https://doi.org/10.1109/SFCS.2001.959942.

[221] Oded Goldreich. Foundations of Cryptography: Volume 1. USA: Cambridge University Press, 2006. ISBN: 0521035368.

[222] Lily Chen et al. Report on post-quantum cryptography. Vol. 12. US Department of Commerce, National Institute of Standards and Technology . . ., 2016. URL: http://dx.doi.org/10.6028/NIST.IR.8105.

[223] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. In: *Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman*. 2022, pp. 365–390. URL: https://doi.org/10.1145/3549993.3550007.

[224] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In: *IEEE Transactions on Information Theory* 31.4 (1985), pp. 469–472. URL: https://doi.org/10.1109/TIT.1985.1057074.

[225] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: *Advances in Cryptology—CRYPTO'98: 18th Annual International Cryptology Conference Santa Barbara, California, USA August 23–27, 1998 Proceedings 18*. Springer. 1998, pp. 13–25. URL: https://doi.org/10.1007/BFb0055717.

[226] Peter L. Montgomery. Modular multiplication without trial division. In: *Mathematics of Computation* 44.170 (1985), pp. 519–521. URL: https://doi.org/10.1090/S0025-5718-1985-0777282-X.

[227] Kenneth Ireland and Michael Ira Rosen. A classical introduction to modern number theory. Vol. 84. Springer Science & Business Media, 1990.

[228] Alfred J. Menezes and Scott A. Van Oorschot Paul C .and Vanstone. Handbook of applied cryptography. CRC Press, 2018.