Title

Interplay between Quantum Computation and Machine Learning

Author

Liao, Haoran

Publication Date

2023

Peer reviewed|Thesis/dissertation

Interplay between Quantum Computation and Machine Learning

by

Haoran Liao

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Physics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor K. Birgitta Whaley, Co-chair
Professor Irfan Siddiqi, Co-chair
Professor Hartmut Häffner
Professor Uroš Seljak

Fall 2023

Interplay between Quantum Computation and Machine Learning

Abstract

Interplay between Quantum Computation and Machine Learning

by

Haoran Liao

Doctor of Philosophy in Physics

University of California, Berkeley

Professor K. Birgitta Whaley, Co-chair

Professor Irfan Siddiqi, Co-chair

Quantum errors remain the primary barrier inhibiting quantum computers from outperforming classical supercomputers. To overcome this challenge, a diverse array of strategies has been developed, encompassing quantum error correction and quantum error mitigation. Machine learning, maturing as a widely adopted approach for pattern recognition, offers new perspectives in enhancing the aforementioned strategies to tackle quantum errors. Furthermore, the implications of quantum errors extend to various applications of quantum computing, notably in quantum machine learning which leverages quantum resources for potential advantage over classical counterparts. This dissertation delves into these intertwined parts, examining the interplay between quantum computation and machine learning. The first part concerns machine learning for enhancing quantum computations. It addresses challenges in correcting errors that occurred to continuously measured logical states, and in improving the efficiency in mitigating errors on both small- and large-scale quantum circuits for increased accuracies in the targeted expectation values, serving as an example of using classical machine learning on quantum data. The second part of this dissertation explores quantum computation for machine learning. It provides theoretical and numerical analysis on the robustness of quantum machine learning models against worst-case errors on input encoded quantum states received through quantum communication, or against quantum decoherence during model training and evaluation, serving as an example of applying quantum machine learning on classical data.

To my parents

Contents

List of Figures

List of Tables

Acknowledgments

I would like to begin by thanking my advisor, Professor K. Birgitta Whaley, for introducing me to the field of quantum computation and information. I am deeply grateful for her guidance, inspiration, and encouragement over the years. She inspired me to become a better scientist and a better communicator, and to stay curious.

I owe a debt of gratitude to William Huggins, a former doctoral student in the Whaley group. He welcomed me into the group and introduced me to the field of quantum machine learning and tensor networks. His generous mentorship helped my early exploration of quantum computing. I also appreciate the support from Ian Convy, a peer doctoral student and close collaborator in the group. I learned a great deal from him in our discussions, which filled much of my research time at Berkeley. I am also grateful to Nam Nguyen and Song Zhang, for introducing me to many aspects of superconducting qubits and being wonderful collaborators. I would also like to thank Zhibo Yang, William Livingston, Philippe Lewalle, and Andrea Rodriguez-Blanco for their solid support of my research at Berkeley. There are other group members and colleagues I would like to thank, including Hang Ren, Ryan Shaffner, Unpil Baek, Tyler Kharazi, Zengzhao Li, Robert Cook, John Paul Marceaux, Akel Hashim, Vincent Su, Kai-Isaak Ellers, and Yulong Dong for their valuable pieces of advice, or helpful discussions on the computers and on the blackboards.

I am grateful to Professor Irfan Siddiqi, Professor Hartmut Häffner, and Professor Uroš Seljak for kindly serving as committee members of my qualifying exam as well as my thesis.

I would also like to express my gratitude to external mentors and collaborators who helped me grow in my career. Alireza Seif and Derek Wang introduced me to quantum error characterization and mitigation, and offered invaluable support to my growth as a scientist. Zlatko Minev taught me much about quantum errors and guided me with patience. Iskandar Sitdikov, Vinay Tripathi, Swarnadeep Majumder, Kunal Sharma, Roland de Putter, and Kevin Sung introduced me to many software and hardware aspects of quantum computing and machine learning that broadened my perspectives.

I would especially like to thank those who supported me to start the journey. Without the encouragement and full support from Professor Sabrina Leslie and Professor Victoria Kaspi, I would not have been able to begin this thesis. Without those who inspired me with their dedication to education like Professor Charles Roth and Professor Robert Littlejohn, I would be less prepared. Without friends along the way, Xuefei Guo, Vincent MacKay, Nathan Leitao, Miles Cranmer, and Qing Shi, I would not have been inspired to continue.

I am also grateful to the friends and companions I have made throughout my time here at Berkeley, Jie Li, Yuehui Lu, Yang Lyu, Tianye Wang, Keming Zhang, Daisong Pan, and Jiayun Wang, who are project teammates, travel buddies, and lunch buddies.

I would also like to express my heartfelt gratitude for the wonderful moments I shared with Yunwen Ji, Chu Fang, Peijie Li, Liang Wu, Grace Harper, Yue Qiu, and others, before, during, and after the pandemic, in California and in New York.

Finally and above all, I give my greatest thanks to my parents Shuling Zhou and Liquan Liao, and my grandmother, for everything.

Part I

Machine Learning for Quantum Information Processing

Chapter 1

Continuous Quantum Error Correction on Superconducting Qubits

This chapter is derived from previously published work by Convy, Liao, Song, Patel, Livingston, Nguyen, Siddiqi, and Whaley [1], where Convy and Liao are the co-first authors, which studied imperfections in continuous quantum error correction measurement signals, proposed a recurrent neural network model for decoding, and benchmarked this machine learning model against the traditional method and a discrete Bayesian model. Liao and Convy are primarily responsible for the implementation of the recurrent neural network model and the discrete Bayesian model, respectively.

1.1 Background on Quantum Error Correction

The prevalence of errors acting upon quantum states, either as a result of imperfect quantum operations or decoherence arising from interactions with the environment, severely limits the implementation of quantum computation on physical qubits. A variety of methods have been proposed to suppress the frequency of these errors, such as dynamic decoupling [2], application of a penalty Hamiltonian [3], and decoherence-free subspace encoding [4]. There also exist various methods for quantum error mitigation [5] which reduce the negative impact of errors on the targeted expectation values. In addition to these tools for error suppression and mitigation, there exist many schemes for quantum error correction (QEC) that are able to return the system to its proper configuration after an error occurs [6]. The ability to correct errors rather than just suppress or mitigate them is vital to the development of fault-tolerant quantum computation [7].

Classically, the most straightforward way to encode information to protect against errors or allow correction of these is to store multiple copies of the same information, namely to create redundancy. In such a way, we could successfully recover the information by a majority vote, shall no more than half of the copies had been corrupted. In quantum mechanics, the No-Cloning theorem states that no physical operations (unitary) can clone an arbitrary

quantum state $|\psi\rangle$. This theorem does not state that we cannot clone a given known state, but it certainly prevents us from making copies of a state in the middle of an arbitrary quantum computation, and thus nullifies the attempt to arbitrarily create redundancy that can assist in protection against corruption of quantum information. The solution to completely protect against errors in principle is the use of quantum error correcting code, where we store each quantum bit in multiple physical qubits in a way that we can detect and correct errors occurring to those physical qubits, enabling us to perform faithful operations on each quantum bit. We call the encoded quantum bit (state) the logical qubit (state), the operator on the encoded state the logical operator, the measurements we performed to detect errors the syndrome measurements, and the operations to reverse the error the recovery operations.

Quantum Error

Quantum error arises when the observed system undergoes undesired interaction with the environment (bath). An error process on the system is formally described as some quantum channel. A quantum channel maps the system density matrix to the reduced density matrix (tracing over the environment) after a joint unitary evolution of the system and environment, namely

$$\mathcal{E}(\rho_S) = \mathrm{Tr}_E[U_{SE}(\rho_S \otimes \rho_E)U_{SE}^\dagger], \tag{1.1}$$

which reflects the fact that the quantum state of the environment degrees of freedom is unmonitored after the joint evolution, such that quantum information of the system has been carried away by the environment. Physically, quantum errors can happen, for example, when we cannot keep track of the quantum state of all the photons that scatter off of the study, we are unaware of the precise duration of the interaction, or how electronic states in an atom are perturbed upon interacting with distant electrical charges [8, 9]. This loss of information is irreversible on the system alone, but can be in principle reversed in an enlarged Hilbert space. This enlarged Hilbert can be simply the original joint system and environment, or a quantum error correcting code encoding the system together with the syndrome measurement ancillas. Assuming the environment initialized to $|0\rangle_E$, Eq. (1.1) can be written as

$$\begin{aligned}\mathcal{E}(\rho_S) &= \sum_i \langle i|_E U_{SE} \left(\sum_{j,k} \rho_{jk} |j\rangle\langle k| \otimes |0\rangle\langle 0|_E \right) U_{SE}^\dagger |i\rangle_E \\ &= \sum_i \langle i|_E U_{SE} |0\rangle_E \left(\sum_{j,k} \rho_{jk} |j\rangle\langle k| \right) \langle 0|_E U_{SE}^\dagger |i\rangle_E .\end{aligned} \tag{1.2}$$

Defining a measurement of the environment performed in the basis $|i\rangle_E$ after the unitary transformation U_{SE} has been applied,

$$E_i \equiv \langle i|_E U_{SE} |0\rangle_E , \tag{1.3}$$

we obtain the operator-sum representation, or Kraus representation, of any quantum channel $\mathcal{E}(\rho_S) = \sum_i E_i \rho_S E_i^\dagger$. It is evident from Eq. (1.3) that $\{E_i\}$ is the set of operators satisfying the completeness condition $\sum_i E_i^\dagger E_i = 1$. It can also be readily shown that any such an operator-sum can be realized by a unitary map[1] acting on an extended system [9] and therefore, this is a bi-directional result.

It can be shown that a completely positive tracing-preserving (CPTP) convex-linear map consists of the axioms defining quantum channels (operations). In fact, a map $\mathcal{E}$ satisfying the CPTP convex-linear axioms if and only if it has an operator-sum representation [8] (again, this is a bi-directional result indicating that any set of operators $\{E_i\}$ satisfying the completeness condition and operates on some density matrix must be a valid, physical operation).

The relationship that a CPTP map implies a unitary map acting on an extended system as in Eq. (1.1) constitutes the application of the Stinespring's dilation theorem [10] in quantum mechanics, which is formally stated as [10]

Theorem 1 . *Any quantum channel, or completely positive and trace-preserving (CPTP) map, $\Lambda : \mathcal{B}(\mathcal{H}_A) \to \mathcal{B}(\mathcal{H}_B)$[2] over finite-dimensional Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$ is equivalent to an isometry (inner-product preserving map) mapping to a higher dimensional Hilbert space $\mathcal{H}_B \otimes \mathcal{H}_E$, where $\mathcal{H}_E$ is also finite-dimensional, followed by a partial tracing over $\mathcal{H}_E$[3] . In particular, the dimension of the ancillary system $\mathcal{H}_E$ can be chosen such that $\dim(\mathcal{H}_E) \leq \dim(\mathcal{H}_A)\dim(\mathcal{H}_B)$ for any Λ.*

We note that the set of Kraus operators of a given error channel is not unique. It is equivalent up to a unitary change of basis, and the number of Kraus operators can be different in another form of the representation. Other representations of error channels are also widely used [11], including the Pauli transfer matrix (PTM) (please see Sec. 2.1 in Chapter 2), χ-matrix, and Choi matrix.

One single-qubit error channel, whose Kraus operators are most easily recognized, is the depolarizing channel

$$\mathcal{E}(\rho) = (1 - \frac{3}{4}p)\rho + \frac{p}{4}(X\rho X + Y\rho Y + Z\rho Z) = (1-p)\,\rho + p\frac{I}{2}. \tag{1.4}$$

Represented as an average over the realizations of a Markov process, i.e., unraveled in quantum trajectories [12], the depolarizing channel on its own describes the statistical ensemble of quantum trajectories subject to a bit flip, a phase flip, or both a bit and a phase flip, each with a $p/4$ probability, equivalent to having a maximally mixed state with probability p. This latter interpretation is used to generalize this error channel to a multi-qubit one. Let Γ be the probability per unit time, or rate, of the depolarizing noise, it can be readily

[1] The unitary map is obtained by further extending the input Hilbert space of an isometry.

[2] We denote the convex set of positive-semidefinite linear operators with unit trace, namely the set of density operators, on a complex Hilbert space $\mathcal{H}$ (thus Hermitian and bounded) as $\mathcal{B}(\mathcal{H})$.

[3] In the Stinespring's representation of such a CPTP map Λ, there exists an isometry $V : \mathcal{B}(\mathcal{H}_A) \to \mathcal{B}(\mathcal{H}_B \otimes \mathcal{H}_E)$ such that $\Lambda(\rho) = \mathrm{Tr}_E(V\rho V^\dagger), \forall \rho \in \mathcal{B}(\mathcal{H}_A)$.

seen that the diagonal elements of the density matrix exponentially decay as

$$e^{-\Gamma t}\rho_{ii} + \frac{1}{2}\left(1 - e^{-\Gamma t}\right), \tag{1.5}$$

leading to an exponential decay of observable expectation values. Although this simple noise model is of theoretical interest due to the fact that all possible single-qubit errors (the three Pauli errors) are included, which fulfills as a test to any error correction model for arbitrary single-qubit error, it is not physically well-motivated.

Decoherence – Relaxation and Dephasing

The more interesting channels are the amplitude damping channel and the dephasing (phase damping, or just phase flip) channel. Let us start with the amplitude damping channel, or energy dissipation. It models physical processes such as spontaneous emission. If the atom starts in the ground state, there is no spontaneous emission and the environment electromagnetic field, starting in no photon, remains in $|0\rangle_E$. If the atom is excited, or in a superposition of ground and excited state, there is a certain probability $p = \Gamma\Delta t$ (the probability per unit time, or rate, Γ of spontaneous emission can be described by Fermi's golden rule) that the excited state of an atom will decay to the ground state and a photon will be emitted. Hence, the environment will make a transition from the state $|0\rangle_E$ (no photon) to the state $|1\rangle_E$ (one photon in some mode). We have the unitary transformation on the atom and the electromagnetic field as

$$\begin{aligned} &|0\rangle_A |0\rangle_E \mapsto |0\rangle_A |0\rangle_E , \\ &|1\rangle_A |0\rangle_E \mapsto \sqrt{p}\,|0\rangle_A |1\rangle_E + \sqrt{(1-p)}\,|1\rangle_A |0\rangle_E . \end{aligned} \tag{1.6}$$

The emitted photon has infinite different modes to propagate into, thus the probability of the atom re-absorbing the photon and re-exciting is negligible. Tracing over the environment, the density matrix of the atom evolves as

$$\rho \mapsto \mathcal{E}(\rho) = \begin{bmatrix} \rho_{00} + p\rho_{11} & \sqrt{1-p}\rho_{01} \\ \sqrt{1-p}\rho_{10} & (1-p)\rho_{11} \end{bmatrix} \tag{1.7}$$

In terms of the decay rate, successfully applying the channel for n times, $(1-p)^n = (1 - \Gamma t/n)^n = e^{-\Gamma t}$, we obtain the system (reduced) density matrix over time as

$$\rho(t) = \begin{bmatrix} \rho_{00} + (1 - e^{-\Gamma t})\rho_{11} & e^{-\frac{\Gamma t}{2}}\rho_{01} \\ e^{-\frac{\Gamma t}{2}}\rho_{10} & e^{-\Gamma t}\rho_{11} \end{bmatrix} \tag{1.8}$$

We call the time for the excited state population to decay to e^{-1} of the initial as T_1 time, and the time for the off-diagonals (the coherence) to decay to e^{-1} of the initial population as T_2 time [9]. With only the amplitude damping channel, we have $T_2 = 2T_1$. Exemplified by

such a channel, any process resulting in the decay of the off-diagonals in the density matrix is decoherence.

Then let us focus on the dephasing channel, the simplest model resulting in decoherence. We first note that coherence, denoting the non-zero off-diagonal elements of a density matrix, arises from a portion of the identically prepared states, in a given basis, having a well-defined relative phase between the constituent superposition states. The phases are what we sometimes refer to as what allows for the wavefunction to interfere in some observable[4]. Decoherence is then a scrambling of the phase information due to undesired environment interaction with the system, such that our knowledge of the system state, reflected in measuring an ensemble of originally identically prepared states, must now be described by a density matrix with decayed off-diagonals. For instance, suppose we start with an ensemble of identically prepared $|+\rangle = 1/\sqrt{2}(|0\rangle + |1\rangle)$, and unmonitored interaction induces a random π-phase flip with $1/2$ probability on each state, our knowledge of the system will decohere to a maximally mixed density matrix of $1/2(|+\rangle\langle+| + |-\rangle\langle-|) = I/2$. More concretely, suppose we have a qubit in $|\psi\rangle = \alpha|+\rangle - \beta|-\rangle$, and we have a $R_x(\pi+\theta)$ gate rotating the qubit around X-axis by π on the Bloch sphere. Suppose the gate has some noise with an over-rotation angle θ which is stochastically sampled from a Gaussian distribution with a mean 0 and a variance 2Γ. The result of this one gate on the qubit, to the best of our knowledge, can only be represented as an average over θ, in a density matrix in the X basis, as follows

$$\rho = \frac{1}{4\pi\lambda}\int_{-\infty}^{\infty} R_x(\pi+\theta)|\psi\rangle\langle\psi|R_x^\dagger(\pi+\theta)e^{-\frac{\theta^2}{4\Gamma}}\,d\theta = \begin{bmatrix} |\alpha|^2 & \alpha\beta^* e^{-\Gamma t} \\ \alpha^*\beta e^{-\Gamma t} & |\beta|^2. \end{bmatrix}. \tag{1.9}$$

The effect of this scramble of the phase shows up as a damping factor in the coherence of the ensemble density matrix. Abstract away, it can be readily verified that the two Kraus operators $E_1 = \sqrt{p}/2(I+Z)$ and $E_2 = \sqrt{p}/2(I-Z)$ describe the dephasing channel

$$\mathcal{E}(\rho) = \begin{bmatrix} \rho_{00} & (1-p)\rho_{01} \\ (1-p)\rho_{10} & \rho_{11} \end{bmatrix} = \begin{bmatrix} \rho_{00} & \rho_{01}e^{-\Gamma t} \\ \rho_{10}e^{-\Gamma t} & \rho_{11} \end{bmatrix}, \tag{1.10}$$

where Γ is the rate of decoherence per unit time. Since this noise channel only affects the coherence, unlike amplitude damping which has an effect on all density matrix elements, it is sometimes emphasized as "pure dephasing".

In physical systems, amplitude damping (Eq. (1.8)) and (pure) dephasing (Eq. (1.10)) occurs simultaneously with two rates, Γ_1, and Γ_ϕ, respectively. Γ_1 is also referred to as the longitudinal relaxation rate. By the definition described previously,

$$\Gamma_1 \equiv \frac{1}{T_1}. \tag{1.11}$$

[4]Consider a superposition state of $1/\sqrt{2}(|0\rangle + |1\rangle)$ whose density matrix in the Z basis have non-zero off-diagonals, when measured in the X basis, the constituent state $|0\rangle = 1/\sqrt{2}(|+\rangle + |-\rangle)$ and $|1\rangle = 1/\sqrt{2}(|+\rangle - |-\rangle)$ constructively interferes to produce a measurement statistics of +1 uniformly. While in an ensemble of fully-decohered states, $1/2(|0\rangle\langle 0| + |1\rangle\langle 1|) = I/2$, each state is unable to interfere and only 1/2 of the states measured in the X basis to have +1.

As shown in (Eq. (1.8)), it also causes the off-diagonal to decay at a rate of $\Gamma_1/2$, which, when combined, leads to an overall decay rate of the off-diagonals $\Gamma_2 = \Gamma_1/2 + \Gamma_\phi$. By the definition described previously,

$$\Gamma_2 \equiv \frac{1}{T_2} = \frac{\Gamma_1}{2} + \Gamma_\phi, \tag{1.12}$$

This is also called the transverse relaxation rate (transverse as on the equatorial plane on the Bloch sphere). It can be measured using e.g., a Ramsey experiment. From Eq. (1.12), it is also evident that when there are both amplitude damping and dephasing, we have

$$T_2 \leq 2T_1. \tag{1.13}$$

So far, we have described incoherent quantum errors, meaning errors that cannot be described by a unitary evolution of the system. What if the over-rotation angle θ of the X gate is deterministic? Then it becomes a coherent error $U_{\mathbf{err}} = e^{-i\theta X}$, when the over-rotation angle is small, expands to $|\psi\rangle \mapsto (1 - i\theta X)|\psi\rangle$, or $\rho \mapsto \rho + \theta^2 X\rho X$, up to a renormalization of the state. Hence, the effect of a small coherent rotation error is the same as a stochastic error with a probability of occurring parametrized by the rotation angle. This is the same as describing what the measurement outcome distribution is after the coherent error evolution.

Quantum Error Correcting Code

Let $\{|\phi_i\rangle\}$ denote the basis set of the code subspace. The quantum error correcting criterion is a sufficient and necessary condition for the recovery operations to be possible [13]. This criterion can be stated based on two principles. First, correctable errors are those that cannot destroy the perfect distinguishability of orthogonal codewords (encoded states). In other words, an error mapping two different encoded states into the same resultant state is surely not correctable since we would not be able to have an inverse mapping from the resultant state back to the desired encoded state. This can be expressed concisely as

$$\langle\phi_i| E_a^\dagger E_b |\phi_j\rangle = C_{abij}\delta_{ij}, \tag{1.14}$$

where the complex tensor C_{abij}, in its full generality, is to be determined. Second, we should not be able to acquire any information about the encoded state by measuring the syndromes, otherwise, we would inevitably disturb that state. This requires that $\langle\phi_i| E_a^\dagger E_b |\phi_i\rangle$ should not depend on i, and thus $C_{abij}\delta_{ij} = C_{ab}\delta_{ij}$. It is then readily true that C_{ab} must be a Hermitian matrix, and we arrive at the QEC criteria,

$$\langle\phi_i| E_a^\dagger E_b |\phi_j\rangle = C_{ab}\delta_{ij}. \tag{1.15}$$

The 3-qubit bit-flip stabilizer code is the quantum analog of classical repetition code. It encodes the logical states $|0\rangle$ and $|1\rangle$ into $|0\rangle_{\mathrm{L}} = |000\rangle$ and $|1\rangle_{\mathrm{L}} = |111\rangle$, respectively, where the stabilizer generators are chosen to be $S_1 = Z_1Z_2$ and $S_2 = Z_2Z_3$, which also serve as the error syndrome operators. The states $|000\rangle$ and $|111\rangle$ span the code subspace, in which

the syndromes have values $(S_1 = +1, S_2 = +1)$. The $(S_1 = -1, S_2 = +1)$, $(S_1 = -1, S_2 = -1)$, $(S_1 = +1, S_2 = -1)$ subspaces are known as the error subspaces, which are spanned by the basis states $\{|011\rangle, |100\rangle\}$, $\{|010\rangle, |101\rangle\}$ and $\{|001\rangle, |110\rangle\}$, respectively. A logical error in quantum memory, i.e., when there is no Hamiltonian evolution, is an error attributed to the logical X operator, $X_L = X_1X_2X_3$. It only protects against weight-one X error, and not any Y or Z error. The same code in the X basis, i.e., $|0\rangle_L = |+++\rangle$ and $|1\rangle_L = |---\rangle$, protects against weight-one Z error, and not any Y or X error.

The 9-qubit Shor's code is a nested version of the above 3-qubit bit-flip code in both the X and Z basis,

$$|0\rangle_L = \left(\frac{1}{\sqrt{2}}|000\rangle + |111\rangle\right)^{\otimes 3} \quad \text{and} \quad |1\rangle_L = \left(\frac{1}{\sqrt{2}}|000\rangle - |111\rangle\right)^{\otimes 3}, \tag{1.16}$$

which is a stabilizer code that can correct any single-qubit error and has a *distance* 3. A phase flip (Z error) in any of the 3-qubit clusters can be detected by measuring

$$X_1X_2X_3X_4X_5X_6 \quad \text{and} \quad X_4X_5X_6X_7X_8X_9. \tag{1.17}$$

A bit-flip (X error) on any of the qubit can be detected by measuring Z_1Z_2, Z_2Z_3 for the first 3-qubit cluster, Z_4Z_5, Z_5Z_6 for the second 3-qubit cluster, and Z_7Z_8, Z_8Z_9 for the third 3-qubit cluster.

We describe a quantum code with n physical qubits, encoding k qubits, and distance $d = 2t + 1$ as an $[[n, k, d]]$ quantum code. We say that a QEC code (QECC) can correct t errors if the set of recoverable errors $\{E_a\}$ includes all Pauli operators of weigh t or less, i.e., the QEC criteria in Eq. (1.15) is satisfied by all Pauli operators $E_a \ E_b$ of weight t or less, where $t = \lfloor (d-1)/2 \rfloor$. A given QECC can detect $d - 1 = 2t$ errors. Therefore, a distance-2 code is useful for detecting weight-1 errors.

Some notable QECCs include the $[[7, 3, 1]]$ Steane code [14], the 5-qubit code which is the smallest possible code protecting one logical qubit against single-qubit errors, i.e., the $[[5, 1, 3]]$ code [15], and Kitaev's surface code $[[2L^2, 2, L]]$ [16, 17].

1.2 Background on Continuous Quantum Error Correction

An essential feature of QEC is the measurement of certain error syndrome operators, which provides information about errors in the physical qubits without collapsing the logical quantum state. In the canonical approach, quantum error correction is conducted in a discrete manner, using quantum logic gates to transfer the qubit information to ancilla qubits and subsequently making projective measurements on these to extract the error syndromes.

In contrast to this theoretical idealization of instantaneous projections of the quantum state, experimental implementation of such measurements inherently involves performing

weak measurements over finite time intervals [18], with the dispersive readouts in superconducting qubit architectures constituting the prime example of this in today's quantum technologies [19, 20, 21, 22]. This has motivated the development of continuous quantum error correction (CQEC) [23, 24, 25, 26, 27, 28, 29, 30, 31, 32], where the error syndrome operators are measured weakly in strength and continuously in time.

CQEC operates by directly coupling the data qubits to continuous readout devices. This avoids the ancilla qubits and periodic entangling gates found in discrete QEC, reducing hardware resources. Additionally, the presence of these entangling gate sequences and ancillas introduces additional error mechanisms, occurring in-between entangling gates or on ancillas, that can cause logical errors [30, 32]. On noisy quantum hardware, multiple rounds of entangling gates and ancilla readouts are required to accurately identify the system state[5]. All of this is also avoided by measuring data qubits directly, as in CQEC.

In addition to quantum memory, CQEC naturally lends itself to modes of quantum computation involving continuous evolution under time-dependent Hamiltonians, such as adiabatic quantum computing [34] and quantum simulation [35]. Given that the Hamiltonians considered generally do not commute with the error operators, the action of an error induces spurious Hamiltonian evolution within the corresponding error subspace until the error is ultimately diagnosed and corrected, resulting in the accrual of logical errors [31]. CQEC can effectively shorten the spurious evolution time in the error subspaces, and therefore increase the target state fidelity in quantum annealing.

Continuous Measurement

A continuous measurement is one in which partial information about the state is extracted continuously in time [18]. The amount of information obtained goes to zero as the duration of the measurement goes to zero. Suppose we want to continuously measure X (Hermitian) with a continuous spectrum of eigenvalues x, and corresponding eigenstates $|x\rangle$. We divide time into intervals of length Δt. In each interval, we have the measurement operator as a Gaussian weighted "sum" of the projectors onto the different eigenstate, centering at α,

$$A(\alpha) \propto \int_{-\infty}^{\infty} e^{[-2k\Delta t(x-\alpha)^2]|x\rangle\langle x|} dx. \tag{1.18}$$

This models a process where there is Gaussian white noise on the measurement eigenvalue $\langle\alpha\rangle = \langle X\rangle$ that the observers observe. To see this more precisely, we compute the probability

[5] In discrete QEC, full syndromes measurements are performed multiple times before attempting to decode, often $\mathcal{O}(n)$ times for a length n repetition code or surface code [33]. This reduces the impact of faulty entangling gates or ancillas.

of having a measurement result of α[6],

$$\begin{aligned} P(\alpha) &= \mathrm{Tr}[A(\alpha)^\dagger A(\alpha)|\psi\rangle\langle\psi|] \propto \int_{-\infty}^{\infty} |\psi(x)|^2 e^{-4k\Delta t(x-\alpha)^2} dx \\ &\approx \int_{-\infty}^{\infty} \Delta(x-\langle X\rangle) e^{-4k\delta t(x-\alpha)^2} dx \propto e^{-4k\Delta t(\alpha-\langle X\rangle)^2}, \end{aligned} \tag{1.19}$$

where at the beginning of the last line we have assumed that when Δt is sufficiently small then the Gaussian is much broader than $\psi(x)$ and so $|\psi(x)|^2$ can be approximated by a delta function centered at the expectation value $\langle X\rangle$.

The action of the measurement operator at each time step on the state $|\psi(t)\rangle$ produces

$$\begin{aligned} |\psi(t+\Delta t)\rangle &\propto A(\alpha)\,|\psi(t)\rangle \propto e^{-2k\Delta t(\alpha-X)^2}\,|\psi(t)\rangle \\ &\propto \{1 - 2k\Delta t X^2 + X[4k\langle X\rangle\Delta t + \sqrt{2k}\Delta W + kX(\Delta W)^2]\}\,|\psi(t)\rangle\,, \end{aligned} \tag{1.20}$$

where we expanded the exponential to first order in Δt (we note that $\Delta W^2 = \Delta t$ in Itô calculus) in the last line.

Taking $\Delta t \to 0$, we set $\Delta t = dt$, $\Delta W = dW$ and $(\Delta W)^2 = dt$, we see that

$$d\,|\psi(t)\rangle \propto \{-[kX^2 - 4kX\langle X\rangle]dt + \sqrt{2k}X dW\}\,|\psi(t)\rangle\,. \tag{1.21}$$

Master Equations

We will give a straightforward "derivation" of the Lindblad master equation and stochastic master equation (SME) arising from the continuous measurement process, as motivated by [18].

By what was described in Sec. 1.1 any transformation of the density matrix must be a CPTP map and it is so if and only if the transformation has an operator-sum representation. Let us ignore the TP (trace-preserving) part for now, which was enforced by the completeness condition of the Kraus operators, and write the most general form of the completely positive transformation as

$$\rho \mapsto \sum_n A_n \rho A_n^\dagger, \tag{1.22}$$

where $\{A_n\}$ are arbitrary operators. We note that under unitary evolution, the Schrödinger equation (a direct consequence of the fundamental postulates of quantum mechanics) says that the density matrix (so it is actually Liouville–von Neumann equation) transforms over a short time interval dt as

$$\rho + d\rho = \rho - \frac{i}{\hbar}[H,\rho]dt = \left(1 - i\frac{H}{\hbar}dt\right)\rho\left(1 + i\frac{H}{\hbar}dt\right). \tag{1.23}$$

where H is the Hamiltonian. Namely, it corresponds to a single infinitesimal transformation operator $A = 1 - iHdt/\hbar$ in Eq. (1.22). Since any complex matrix can be expressed in terms

[6]Note that the probability of obtaining measurement result m of a positive operator-valued measure (POVM) Ω_m is $P(m) = \mathrm{Tr}[\Omega_m \rho \Omega_m^\dagger]$, and the resultant density matrix is $\rho_f = \Omega_m \rho \Omega_m^\dagger / \mathrm{Tr}[\Omega_m \rho \Omega_m^\dagger]$.

of Hermitian and anti-Hermitian matrices, the arbitrary operator A can be further expressed as $A = 1 - iHdt/\hbar + bdt$, where b is Hermitian and all the anti-Hermitian part goes into $iH/\hbar$. It turns out that the (continuous) measurement process brings another contribution to A in terms of the Wiener differential dW, such that $dW^2 = dt$ in Itô calculus, as shown in Eq. (1.21). Therefore, we should include the possible dW term in the expression of A,

$$A = 1 - i\frac{H}{\hbar}dt + bdt + cdW, \tag{1.24}$$

where c is an arbitrary operator.

Now, by the operator sum in Eq. (1.22),

$$d\rho = -\frac{i}{\hbar}[H,\rho]dt + \{b,\rho\}dt + c\rho c^\dagger dt + (c\rho + \rho c^\dagger)dW. \tag{1.25}$$

We can then average over all possible Wiener processes (all possible measurement records), or discard the measurement records, which is denoted by taking $\langle\!\langle\cdot\rangle\!\rangle$. Noting that $\langle\!\langle \rho dW\rangle\!\rangle$=0 in Itô calculus,

$$d\langle\!\langle\rho\rangle\!\rangle = -\frac{i}{\hbar}[H,\langle\!\langle\rho\rangle\!\rangle]dt + \{b,\langle\!\langle\rho\rangle\!\rangle\}dt + c\langle\!\langle\rho\rangle\!\rangle c^\dagger dt. \tag{1.26}$$

Since $\langle\!\langle\rho\rangle\!\rangle$ is an average over valid density matries, it is also a valid density matrix and must have unit trace, or $d\,\mathrm{Tr}[\langle\!\langle\rho\rangle\!\rangle] = \mathrm{Tr}[d\langle\!\langle\rho\rangle\!\rangle] = \mathrm{Tr}[\langle\!\langle\rho\rangle\!\rangle(2b + c^\dagger c)] = 0$. For this to hold for an arbitrary density matrix, it must be that $b = -(c^\dagger c)/2$. And we arrive at the Lindblad master equation:

$$d\rho = -\frac{i}{\hbar}[H,\rho]dt + \mathcal{D}[c]\rho dt, \tag{1.27}$$

where

$$\mathcal{D}[c]\rho \equiv c\rho c^\dagger - \frac{1}{2}(c^\dagger c\rho + \rho c^\dagger c) \tag{1.28}$$

is the Lindblard superoperator. The Lindblad master equation is deterministic, as it represents an "averaging out" of the system-environment interaction, that it averaged over all possible noise realizations. Its formal derivation is achieved by performing a partial trace over the environment following a unitary system-environment interaction, assuming Born approximation and Markov approximation [36]. A simple example is the Hamiltonian $H = Z$ with the Lindbladian $L = \sqrt{\gamma}\sigma_-$ that corresponds to the amplitude damping channel with a spontaneous emission rate of γ.

The full transformation without averaging over the noise realizations (measurement records) becomes

$$d\rho = -\frac{i}{\hbar}[H,\rho]dt + \mathcal{D}[c]\rho dt + (c\rho + \rho c^\dagger)dW. \tag{1.29}$$

To preserve the trace of the density operator, we need to add a term $\rho\,\mathrm{Tr}[\rho(c + c^\dagger)dW]$ so

$$d\rho = -\frac{i}{\hbar}[H,\rho]dt + \mathcal{D}[c]\rho dt + (c\rho + \rho c^\dagger)dW - \rho\,\mathrm{Tr}[\rho(c + c^\dagger)]dW. \tag{1.30}$$

which enforces that $\mathrm{Tr}[d\rho] = \mathrm{Tr}[\rho(c+c^\dagger)dW] - \mathrm{Tr}\{\rho\,\mathrm{Tr}[\rho(c+c^\dagger)]dW\} = \{\mathrm{Tr}[\rho(c+c^\dagger)] - \mathrm{Tr}[\rho(c+c^\dagger)]\,\mathrm{Tr}(\rho)\}dW = 0$. Therefore, we arrive at the stochastic master equation (SME):

$$d\rho = -\frac{i}{\hbar}[H,\rho]dt + \mathcal{D}[c]\rho dt + \mathcal{H}[c]\rho dW, \tag{1.31}$$

where

$$\mathcal{H}[c]\rho \equiv c\rho + \rho c^\dagger - \rho\langle c + c^\dagger\rangle_\rho \tag{1.32}$$

is the measurement superoperator.

Further taking into account multiple measurement operators c_n with separate Wiener noise process dW_n independent of all the others, as well as measurement efficiencies η_n, we obtain a more general SME

$$d\rho = -\frac{i}{\hbar}[H,\rho]dt + \sum_n \left(\mathcal{D}[c_n]\rho dt + \sqrt{\eta_n}\mathcal{H}[c_n]\rho dW_n\right), \tag{1.33}$$

whose corresponding measurement record for the n-th process can be written as

$$dr(t) = \frac{c_n + c_n^\dagger}{2}dt + \frac{1}{\sqrt{4\eta_n}}dW. \tag{1.34}$$

The measurement superoperator $\mathcal{H}[c]\rho$ represents the information gain due to the measurement process (if we average over, or discard, the measurement records, then this term does not exist) that it modifies the state depending on the measurement records. The $\mathcal{D}[c]\rho$ term represents the back-action or the disturbance to the state due to the measurement independent of whether or not the measurement records are used by the observers.

Homodyne Measurements

Homodyne measurement is a technique used in quantum optics and quantum information processing. It involves the mixing of a quantum signal with a strong reference beam, known as a local oscillator, at a beam-splitter. The relative phase between the signal and the local oscillator is crucial. By adjusting this phase, one can measure different quadratures of the quantum field.

The interaction Hamiltonian for the transmission line and the cavity field is given by a Jaynes-Cummings Hamiltonian

$$H = i\sqrt{\frac{\gamma}{\Delta t}}(ba^\dagger - b^\dagger a), \tag{1.35}$$

where γ is the coupling strength and Δt is some coarse-grained time-scale in the collision model (see Eq. (14, 16, 17) in [37]), b and a are the lowering operators of the cavity field and the transmission line, respectively.

The original Hamiltonian in Eq. (1.35) then generates a unitary which we keep up to order Δt:

$$U = e^{-iH\Delta t} \approx 1 + \sqrt{\gamma\Delta t}(ba^\dagger - b^\dagger a) - \frac{\gamma}{2}(ba^\dagger - b^\dagger a)^2\Delta t. \tag{1.36}$$

The homodyne measurement readouts the quadrature basis of the probe, in-phase I, quadrature Q, or some linear combination thereof, and can be implemented by a variety of devices. In our physical experiments, we use JPAs. For our analysis, we will measure in the I quadrature, in which we construct the quadrature operator $R = a + a^\dagger$. Measuring in this basis, the output is a continuous variable r with associated Kraus operators [38] (also see Eq. (1.3) in Chapter 1)

$$\begin{aligned}\Omega_r &= \langle r|U|0\rangle \\ &= \langle r|0\rangle + \langle r|1\rangle\sqrt{\gamma\Delta t}b - \frac{\gamma}{2}\Delta t\left(\langle r|0\rangle b^\dagger b + \langle r|2\rangle\sqrt{2}b^2\right) \\ &= \langle r|0\rangle\left[1 + r\sqrt{\gamma\Delta t}b - \frac{\gamma}{2}\Delta t(b^\dagger b - (r^2-1)b^2)\right],\end{aligned} \tag{1.37}$$

where $\langle r|0\rangle = (2\pi)^{-1/4}\exp(-r^2/4) = \sqrt{P_0(r)}$ is the probes's ground state in the position basis and $P_0(r)$ is the probability of measuring r when the probe is in the ground state. In the last line, we have used the Hermite polynomials to express the harmonic oscillator's first and second excited states in terms of its ground state.

We determine the probability of measuring a particular outcome r as

$$p_r = \langle\Omega_r^\dagger\Omega_r\rangle_\rho = P_0(r)\left[1 + r\sqrt{\gamma\Delta t}\,\langle b + b^\dagger\rangle_\rho + \gamma\Delta t(r^2-1)\,\langle b^\dagger b\rangle_\rho\right], \tag{1.38}$$

where the average is taken over the states ρ of the cavity field coupled to the transmons [39].

If we approximate r as a Gaussian variable, we then want to determine the mean and variance of this:

$$\begin{aligned}\langle r\rangle_\rho &= \int_{-\infty}^{\infty} r p_r dr = \sqrt{\gamma\Delta t}\,\langle b + b^\dagger\rangle_\rho, \\ \langle r^2\rangle_\rho &= \int_{-\infty}^{\infty} r^2 p_r dr = 1.\end{aligned} \tag{1.39}$$

Let ΔW be drawn from a Gaussian distribution with variance Δt. The statistics of the measurement record of r can be reproduced by

$$r\sqrt{\Delta t} = \sqrt{\gamma}\,\langle b + b^\dagger\rangle_\rho\,\Delta t + \Delta W. \tag{1.40}$$

The voltage operator to be measured will be of the form

$$\hat{V} \propto \frac{a + a^\dagger}{\sqrt{\Delta t}}, \tag{1.41}$$

resulting in a classical voltage

$$V = A\frac{r}{\sqrt{\Delta t}}, \tag{1.42}$$

where A is a constant scaling factor in units of $\mathrm{V}\cdot s^{1/2}$ characterising the physical noise power in a certain bandwidth. Using Eq. (1.40), the measured voltage V, which is written in terms of

$$V\Delta t = A\left(\sqrt{\gamma}\left\langle b + b^{\dagger}\right\rangle_{\rho}\Delta t + \Delta W\right), \tag{1.43}$$

has variance that scales as Δt^{-1}. The state of the transmons can be inferred from the homodyne measurement voltage in Eq. (1.43) [39].

To implement a single parity measurement on two qubits, we dispersively couple two qubits to the same readout resonator. We tune the qubits to have the same dispersive coupling to the resonator so that the states $|01\rangle$ and $|10\rangle$ are indistinguishable on the I-Q plane. By making the dispersive shift χ much larger than the linewidth κ of the resonator, we can make the reflected phase of $|00\rangle$ (close to π) and $|11\rangle$ (close to $-\pi$) overlap with one another, making them indistinguishable as well. Altogether we implement a full parity measurement of odd excitations vs. even excitations by measuring the I quadrature. In our experiment, we implement two of these full parity measurements – one between qubits 1 and 2 and the other between qubits 2 and 3 [32].

1.3 Continuous Quantum Error Correction on Small Stabilizer Code

Motivation of Applying Machine Learning

Previous theoretical work on CQEC has focused primarily on measurement signals that behave in an idealized manner [30, 31, 29], such that each sample is assumed to be i.i.d. Gaussian (white noise) with a mean given by one of the syndrome eigenvalues. The presence of white noise requires some inference of the measured state, which is a key component in the decoding of CQEC schemes. However, in real dispersive readout signals, we observe a wide variety of "imperfections" caused by hardware limitations and post-processing effects, which can lead to more complicated syndrome dynamics or significant alterations to the noise distribution. A well-calibrated CQEC protocol should be designed to take into account any significant non-ideal behavior for a given architecture in addition to dealing with the continuous nature of the signal. However, it is often difficult to generate a precise mathematical description of the imperfections present in real measurement signals.

Machine learning algorithms offer a solution to this problem, as they can be optimized to solve a task by looking directly at the relevant data instead of relying on hard-coded decision rules [40]. Highly expressive models involving multiple neural network layers have proven to be particularly effective at solving complex tasks such as image recognition and language translation [41]. The recurrent neural network (RNN) is a popular sequential learning model, because it operates on inputs of varying length and provides an output at each step. After being trained on a set of non-ideal measurement signals, an RNN can function as a CQEC algorithm by generating probabilities that describe the likelihood of an error at a given time

step. Most importantly, the flexibility of the algorithm allows it to handle imperfections in the signal that would otherwise be impractical to model.

Here we investigate the performance of an RNN-based CQEC algorithm which acts on measurement signals with non-ideal behavior. We emphasize here *active correction*, in which errors are corrected during the experiment as soon as they are observed. To quantify the benefits of using a neural network, we compare the RNN to a conventional double threshold scheme as well as to a discrete Bayesian classifier. The first threshold scheme for CQEC was by Sarovar *et al.* [26], who used the sign of the averaged measurement signals (i.e., a threshold at zero) to identify the error subspace. This filter was improved upon in Atalaya *et al.* [29] and Atalaya, Zhang *et al.* [31], as well as in Mohseninia *et al.* [30], by adding a second threshold to better detect errors that affect multiple syndromes. We chose to compare our RNN model to the threshold scheme in [31], since it had superior performance in numerical tests.

Problem Setup

We exemplify our CQEC protocol by operating it on the three-qubit bit-flip stabilizer code; in general, the protocol works with any QEC code. This is a proof-of-concept of the continuous QEC scheme with a machine-learning approach for signal processing, which is an extension of the concept of decoder into the continuous quantum error correction regime.

In the continuous operation of the three-qubit bit-flip code, the error syndrome operators $S_k, k = \{1, 2\}$ are continuously and simultaneously measured to yield the following idealized signals for each S_k as a function of time t as in Eq. (1.34):

$$I_k(t) = \sqrt{\Gamma_m^k}\,\mathrm{tr}[S_k\rho(t)] + \xi_k(t). \tag{1.44}$$

Here $\rho(t)$ is the density matrix of the three physical qubits and Γ_m^k is the measurement strength that determines the time to sufficiently resolve the mean values of the syndromes under constant variance. Specifically, $1/\Gamma_m^k$ is the time needed to distinguish between the eigenvalues of S_k with a signal-to-noise ratio (SNR) of 1[7]. In the Markovian approximation, $\xi_k(t)$ is Gaussian white noise, i.e., $\xi(t) = \dot{W}(t)$ where $W(t)$ is a Wiener process, with a two-time correlation function $\langle \xi_k(t)\xi_{k'}(t')\rangle = \delta_{kk'}\delta(t-t')$, where the $\langle\cdot\rangle$ denotes average over an ensemble of noise realizations. In the continuous operation, the observer receives noisy voltage traces with means proportional to the syndrome operator eigenvalues and variances that determine the continuous measurement collapse timescales. Monitoring both error syndromes with streams of noisy signals represents a gradual gain of knowledge of the measurement outcome to diagnose bit-flip errors that occur. We shall refer to the parity of $I_k(t)$ as even or odd depending on whether the mean value of $I_k(t)$ is positive or negative. In an actual experiment, we will only have access to the averaged signals taken at discrete

[7]The SNR is defined as $(\mu_e - \mu_o)^2/(\sigma_e + \sigma_o)^2$, where μ and σ are the mean and standard deviation of the signals, and subscripts denote the odd and even parities of the syndrome measurements.

time steps separated by Δt, which we denote by $I_{k,t}$ at time step t:

$$I_{k,t} = \sqrt{\Gamma_m^k}\,\mathrm{tr}\left[S_k\rho(t)\right] + \frac{\Delta W}{\Delta t} \tag{1.45}$$

where $\Delta W \sim \mathcal{N}(0, \Delta t)$. We shall assume that $\rho(t)$ only changes due to bit-flips at the beginning of each time step Δt for very small Δt.

In previous work, Ref. [30] compared the performance of a linear approximate Bayesian classifier and the double threshold model with one threshold fixed at $y = 0$ and another threshold at $y > 0$ in correcting the three-qubit bit-flip code for quantum memory. Ref. [31] analyzed the double threshold model with two varying thresholds in correcting the three-qubit bit-flip code, and applied it to quantum annealing under bit-flip errors X_q with which the chosen annealing Hamiltonian does not commute. In the current work, we shall study the performance of machine learning algorithms both in quantum memory and in quantum annealing.

The SME governing the evolution of $\rho(t)$ under measurements with a finite rate of information extraction implied by Eq. (1.44) in the presence of bit-flip errors is given by [26, 31]

$$\begin{aligned}\dot{\rho}(t) = &-i[H(t),\rho] \\ &+ \sum_{k=1,2}\left[\frac{\Gamma_\phi^k}{2}\left(S_k\rho S_k - \rho\right) + \sqrt{\Gamma_m^k}\xi_k(t)\left(\frac{S_k\rho + \rho S_k}{2} - \rho\left\langle S_k\right\rangle_\rho\right)\right] + \sum_{q=1,2,3}\gamma_q\left(X_q\rho X_q - \rho\right).\end{aligned} \tag{1.46}$$

The first term describes the coherent evolution of the three-qubit state under a Hamiltonian $H(t)$, which can, for instance, be a quantum annealing Hamiltonian. The second term describes the back-action induced by the simultaneous continuous measurement of the error syndrome operators S_1 and S_2 on the three-qubit state, where Γ_ϕ^k is the measurement-induced ensemble dephasing rate of the corresponding error syndrome operator S_k. The measurement strength Γ_m^k, is related to the detector efficiency η_k as $\Gamma_m^k = 2\Gamma_\phi^k\eta_k$ The first two terms can be obtained by substituting operators $c_k \propto S_k$ into the general SME $d\rho = -i[H,\rho]dt + \sum_k(\mathcal{D}[c_k]\rho dt + \sqrt{\eta_k}\mathcal{H}[c_k]\rho dW)$ as derived in Eq. (1.33). The third term describes the decoherence of the three-qubit state in the presence of bit-flip errors, with $\gamma_q, q = \{1,2,3\}$ denoting the bit-flip error rate of the q^{th} physical qubit. While the idealized measurement signals mentioned above assume no effect induced by physical experimental apparatus in the qubit readouts, there are various imperfections of the measurement signals in practice that make the error diagnosis more challenging. We shall first present the characteristics of these measurement signals from physical experiments below and explain their implications for our purpose.

Characteristics of CQEC Measurement Signals

The superconducting qubits are monitored using voltage signals from homodyne measurements of the parity operators that are derived from tones reflected off the resonator (see

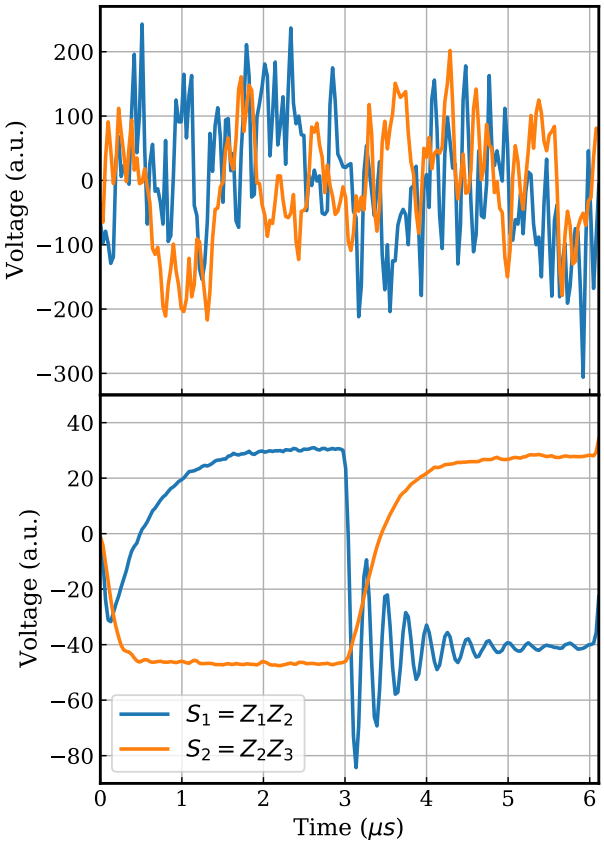


Figure 1.1: The measurement signals of the two syndrome operators $S_1 = Z_1Z_2$ and $S_2 = Z_2Z_3$ on the transmon qubits. The even(odd) parity signal, i.e., $S_k = +1(-1)$ has a voltage readout that is centered at an arbitrary negative(positive) value, according to Eq. (1.43). We note that the experimental voltage readout of even parity is centered at the negative mean by design. The upper figure is the raw voltage signal readout of a single experimental run. The lower figure is the averaged voltage readout over $47,494$ post-selected runs. The qubits are initialized to $|100\rangle$ and an X_2 bit-flip is artificially injected at $t = 3.0\,\mu s$, resulting in a new state $|110\rangle$. The oscillation pattern is explained in Sec. 1.3.

Sec. 1.2). The resonator signal is fed into a Josephson parametric amplifier (JPA) in order to increase the signal strength without adding a significant amount of noise. The amplified radio frequency signals are then demodulated and digitized. After a further digital demodulation, the signals are processed with an exponential anti-aliasing filter with a time constant of 32 ns. This filtered signal, which is averaged in $\Delta t = 32\,\text{ns}$ bins, is then streamed from the digitizer card to the computer.

Due to the effects of the amplifier and resonator, we expect that measurements performed on such real superconducting devices will deviate from the idealized behavior predicted by Eq. (1.44). In particular, we can anticipate the following three imperfections:

1. The noise will possess a high degree of positive auto-correlation at short temporal lags due to the narrow low-pass bandwidth of the JPA and anti-aliasing filter.

2. When a bit-flip occurs, the syndrome means will change gradually rather than instantaneously as the resonator reaches its new steady state. These periods are referred to as *resonator transients* to stress their temporary nature, and arise because of time-dependent changes in the measurement strength Γ_m^k (see Sec. 1.3).

3. The values of the syndromes will drift over time due to small changes in experimental conditions (e.g. temperature). Unlike the other imperfections, this effect is only noticeable when comparing *across* quantum trajectories rather than within them.

These non-ideal behaviors in the measurement signals extracted from our typical physical experiments will be incorporated into our simulated experiments in Sec. 1.5.

Fig. 1.1 shows experimental dispersive readouts taken from three transmon qubits [42] over the span of 6 µs [32]. The blue and orange lines are a record of the outputs from the two resonators, each measuring a different pair of qubits for their syndromes. The top figure shows the measurement signals from a single experiment, which contains large amounts of auto-correlated noise. During the experiment, an X_2 error was injected at 3.0 µs, flipping the system from $|100\rangle$ to $|110\rangle$, but the weak-measurement noise largely obscures its effect on the syndrome values.

To reveal these underlying syndromes, the bottom figure of Fig. 1.1 shows an average over the measurements from roughly 47,500 experiments, each initialized to $|100\rangle$ and injected with an X_2 error at 3.0 µs. It takes approximately 2 µs after initialization for the syndromes to reach their steady-state values for $|100\rangle$, as the number of photons in each resonator increases from zero gradually. We ignore this effect in our analysis, as it will only occur once at the start of an experiment. After the X_2 error is injected, the syndromes do not instantaneously jump to a new pair of values but instead enter a transitory period which can include significant oscillations. These transients derive from the time-dependent changes in the measurement rate $\Gamma_m^k(t)$ analyzed in Sec. 1.3. This period lasts for roughly 2 µs, after which the syndromes stabilize at their new steady-state values for $|110\rangle$.

Depending on the underlying hardware, a measurement signal may be generated on a wide variety of different scales, such as the arbitrary voltage scale in Fig. 1.1. To denote a

signal generically on any scale, we write the measurement samples as

$$I_{k,t} = \bar{S}_{k,t} + \sqrt{\tau_k}\varepsilon_t, \tag{1.47}$$

where $\bar{S}_{k,t}$ is the scaled mean of the k-th resonator at step t, τ_k is the scaled variance of the k-th resonator, and $\varepsilon_t \sim \mathcal{N}(0,1)$. In this notation, the physical quantities Γ_m and Δt from Eq. (1.45) have been absorbed into $\bar{S}_{k,t}$ and τ_k.

Resonator Transients

The resonator transients are manifested from the varying SNR before the qubit-state-dependent coherent states $|\alpha_{\zeta\eta}(t)\rangle$ of the microwave field in the cavity reach their steady states when the resonator linewidth κ is small, where $\zeta, \eta \in \{e, g\}$ and e/g denotes the excited/ground state. The complex field amplitude $\langle \hat{a} \rangle_{\zeta\eta} = \alpha_{\zeta\eta}$ given that the qubits are in state $\zeta\eta$ satisfies [39, 43, 20]

$$\begin{cases} \dot{\alpha}_{ee}(t) = -i\varepsilon - i(\delta_r + 2\chi)\alpha_{ee}(t) - \frac{\kappa}{2}\alpha_{ee}(t), \\ \dot{\alpha}_{gg}(t) = -i\varepsilon - i(\delta_r - 2\chi)\alpha_{gg}(t) - \frac{\kappa}{2}\alpha_{gg}(t), \\ \dot{\alpha}_{eg}(t) = -i\varepsilon - i\delta_r\alpha_{eg}(t) - \frac{\kappa}{2}\alpha_{eg}(t), \\ \dot{\alpha}_{ge}(t) = -i\varepsilon - i\delta_r\alpha_{ge}(t) - \frac{\kappa}{2}\alpha_{ge}(t), \end{cases} \tag{1.48}$$

where ε is the amplitude of the driving tone, χ is the dispersive shift and $\delta_r = \omega_r - \omega_d$ is the detuning of the measurement drive to the bare cavity frequency.

The steady state ($\dot{\alpha}_{\zeta\eta} = 0$) solutions to the above equations are

$$\begin{cases} \alpha_{ee/gg} = \frac{-2\varepsilon}{2(\delta_r \pm 2\chi) - i\kappa}, \\ \alpha_{eg} = \alpha_{ge} = \frac{-2\varepsilon}{2\delta_r - i\kappa} \end{cases} \tag{1.49}$$

with + for ee and − for gg.

In our parity measurement, we probe at the shared odd excitation resonance, which is also the same as the bare cavity frequency, i.e., $\delta_r = 0$. The cavity resonance when the qubits are in $|11\rangle$ is shifted from the bare cavity resonance by $2\chi/2\pi = -4\,\mathrm{MHz}$, while the resonance when the qubits are in $|00\rangle$ is shifted from the bare frequency by $-2\chi/2\pi = 4\,\mathrm{MHz}$. This results in an asymmetry between the paths in phase space leading up to the steady states when the qubit pair changes parity.

When the qubits go from an even-parity state to an odd-parity state, e.g., $|00\rangle \to |10\rangle$, solving $\dot{\alpha}_{eg}(t)$ in Eq. (1.48) with the initial coherent state at α_{gg} yields the path $\alpha_{eg}(t)$ specified by

$$\begin{cases} \alpha_{gg}(t) = \alpha_{gg} \\ \alpha_{eg}(t) = \left(\alpha_{gg} + \frac{2i\varepsilon}{\kappa + 2i\delta_r}\right) e^{-i\delta_r t - \frac{\kappa}{2}t} - \frac{2i\varepsilon}{\kappa + 2i\delta_r}. \end{cases} \tag{1.50}$$

When the qubits go from an odd-parity state to an even-parity state, e.g., $|10\rangle \to |00\rangle$, solving $\dot{\alpha}_{gg}(t)$ in Eq. (1.48) with the initial coherent state at α_{gg} yields the path $\alpha_{gg}(t)$ specified by

$$\begin{cases} \alpha_{gg}(t) = \left(\alpha_{eg} + \frac{2i\varepsilon}{\kappa+2i(\delta_r-2\chi)}\right) e^{-i(\delta_r-2\chi)t-\frac{\kappa}{2}t} - \frac{2i\varepsilon}{\kappa+2i(\delta_r-2\chi)} \\ \alpha_{eg}(t) = \alpha_{eg}. \end{cases} \tag{1.51}$$

These paths are shown in Fig. 1.2. Strictly speaking, the two sets of solutions apply when there are no dynamics apart from the dispersive measurements.

The measurement strength is defined as [44, 39]

$$\Gamma(t) = \frac{1}{2}\kappa|\alpha_{gg}(t) - \alpha_{eg}(t)|^2, \tag{1.52}$$

which scales the separation of the two parity signal means under constant noise variance (see Eq. (1.44)). In the odd-to-even parity transition, the path in phase-space leading up to the steady states forms a tighter spiral as the ratio $|\chi/\kappa|$ gets larger. A tighter spiral translates to a more oscillatory $\Gamma(t)$, thus leading to a more oscillatory signal mean [20].

Shown in Fig. 1.3, the ring-up transient without clear oscillations is manifested in the measurement strength corresponding to the even-to-odd parity transition in Eq. (1.50), whereas the ring-down transient with oscillations is manifested in the measurement strength corresponding to the odd-to-even parity transition in Eq. (1.51). They show good agreement with experimental observations, such as those in Fig. 1.1.

Impact of Auto-correlations

Unlike the other imperfections, the challenge posed by auto-correlated signal noise can be characterized theoretically. If the Gaussian noise in $I_{k,t}$ is correlated, then the distribution of noise samples can be parameterized in terms of a covariance matrix Σ whose off-diagonal elements determine the degree of correlation. For simplicity, we restrict our analysis to dependencies that are Markovian, such that $I_{k,t}$ depends only on the preceding measurement $I_{k,t-1}$, though our conclusions are not limited to this regime. Using a correlation coefficient of $0 < \rho < 1$, the joint Gaussian log-density describing $I_{k,t}$ and $I_{k,t-1}$ is

$$\log\ p(I_{k,t}I_{k,t-1}|\bar{S}_{k,t}) = -\frac{1}{2\tau_k(1-\rho^2)}\begin{bmatrix}\tilde{I}_{k,t} & \tilde{I}_{k,t-1}\end{bmatrix}\begin{bmatrix}1 & -\rho \\ -\rho & 1\end{bmatrix}\begin{bmatrix}\tilde{I}_{k,t} \\ \tilde{I}_{k,t-1}\end{bmatrix} + A, \tag{1.53}$$

where $\tilde{I}_{k,j} \equiv I_{k,j} - \bar{S}_{k,j}$ denotes the centered signal sample at step j and A is the log of the normalization constant. We shall assume hereafter that the signal has been rescaled such that $\bar{S}_{k,j} = \pm 1$.

The effect of auto-correlations on error correction is best characterized in terms of how it impacts the usefulness of the syndrome measurements. To be more precise, we know that the purpose of each measurement is to provide some information about whether the underlying syndrome value of the state is 1 or -1. When framed in these terms, we can formalize and

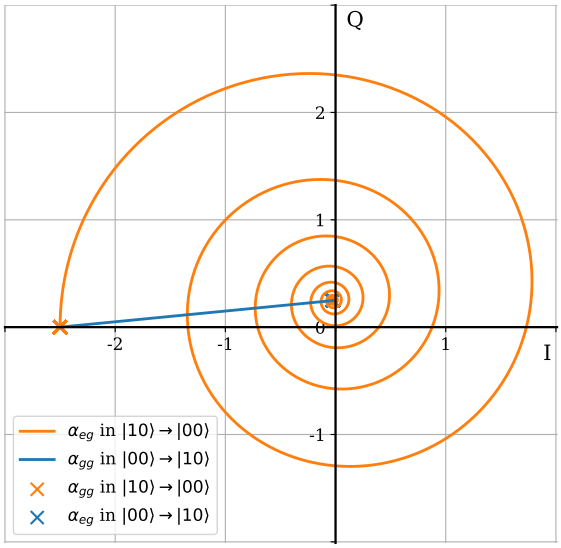


Figure 1.2: The pointer state paths leading up to the steady state in the phase space, with $\kappa/2\pi = 800\,\text{kHz}$, $\chi/2\pi = -2\,\text{MHz}$, $\delta_r = 0$ and ε set to 1. When the qubit pair goes from an even parity to an odd parity, e.g., $|00\rangle \to |10\rangle$, the blue line is the path of $\alpha_{eg}(t)$ while the blue cross shows the steady state of α_{gg}, obtained from Eq. (1.50). When the qubit pair goes from an odd parity to an even parity, e.g., $|10\rangle \to |00\rangle$, the orange spiral curve is the path of α_{gg} while the orange cross shows the steady state of α_{eg}, obtained from Eq. (1.51).

quantify a notion of measurement "usefulness" using Bayesian theory, specifically a ratio called the *Bayes factor* which we denote as ϕ [45]. This factor can be written in log form as

$$\log \phi_{k,t} = \log p(I_{k,t}|I_{k,t-1}, \bar{S}_{k,t} = 1) - \log p(I_{k,t}|I_{k,t-1}, \bar{S}_{k,t} = -1), \tag{1.54}$$

and quantifies how much evidence $I_{k,t}$ gives about the underlying syndrome value if we have already seen the previous measurement $I_{k,t-1}$. The larger the magnitude of $\log \phi_{k,t}$ the more useful $I_{k,t}$ is for our task, with its sign simply indicating whether the evidence supports a value of 1 or -1.

Let $Q = \Sigma^{-1}$. By making the substitutions $\sigma^{-1} = Q_{22}$ and $\mu = \bar{S}_{k,t} - Q_{12}/Q_{22}(I_{k,t-1} - \bar{S}_{k,t})$ in the unconditional log-densities $-(I_{k,t} - \mu)^2/(2\sigma) + A$, each of the conditional log-densities in Eq. (1.54) can be written as

$$\log p(I_{k,t}|I_{k,t-1}, \bar{S}_{k,t}) = -\frac{[I_{k,t} - \bar{S}_{k,t} - \rho(I_{k,t-1} - \bar{S}_{k,t})]^2}{2\tau_k(1-\rho^2)} + A, \tag{1.55}$$

where A is again the normalization constant [46]. Expanding the numerator and keeping only the terms that depend on $\bar{S}_{k,t}$ gives

$$\log p(I_{k,t}|I_{k,t-1}, \bar{S}_{k,t}) \to \frac{S_k^2(\rho-1) + 2\bar{S}_{k,t}(I_{k,t} - \rho I_{k,t-1})}{2\tau_k(1+\rho)}, \tag{1.56}$$

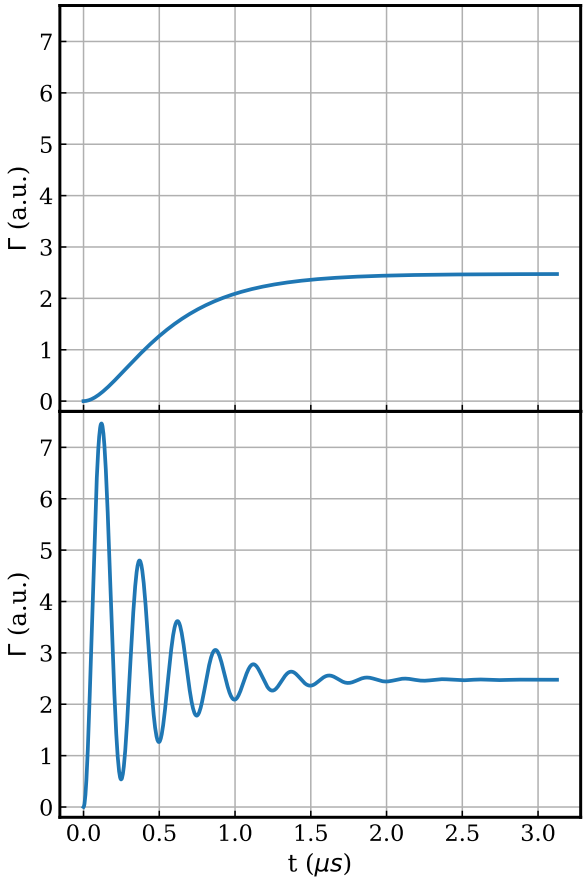


Figure 1.3: The measurement rate $\Gamma(t)$ on a pair of qubits with a bit-flip transition at $t = 0$, with $\kappa/2\pi = 800\,\text{kHz}$, $\chi/2\pi = -2\,\text{MHz}$, $\delta_r = 0$ and ε set to 1. The upper figure corresponds to the qubit pair transitioning from an even parity to an odd parity, obtained from Eq. (1.50). The lower figure corresponds to the qubit pair transitioning from an odd parity to an even parity, obtained from Eq. (1.51).

where we ignore the other terms since they will cancel when computing $\log \phi_{k,t}$. After substituting this representation back into Eq. (1.54) we get

$$\log \phi_{k,t} = \frac{2(I_{k,t} - \rho I_{k,t-1})}{\tau_k(1+\rho)}, \tag{1.57}$$

where the value of $\log \phi_{k,t}$ depends not only on $I_{k,t}$ and $I_{k,t-1}$ but also on the variance and auto-correlation of the measurements.

To see the impact of the auto-covariance more clearly, we compute the expectation value $\mathbb{E}[\log \phi_{k,t}]$ with respect to a Gaussian distribution centered on the true syndrome value $S'_{k,t} = \pm 1$. Since Eq. (1.57) is linear, we can simply substitute in $S'_{k,t}$ for $I_{k,t}$ and $I_{k,t-1}$ to get $\mathbb{E}[\log \phi_{k,t}]$. After taking its magnitude, we have

$$|\mathbb{E}[\log \phi_{k,t}]| = \frac{2(1-\rho)}{\tau_k(1+\rho)}, \tag{1.58}$$

which decreases as the value of ρ increases. Eq. (1.58) shows that positive auto-correlation ($\rho > 0$) in the signal makes each of our measurements less useful than if the noise had been uncorrelated ($\rho = 0$), which means that it will take longer for us to determine the value of $\bar{S}_{k,t}$ at a given measurement strength.

This result can be understood by imagining that $\bar{S}_{k,t}$ and $I_{k,t-1}$ are competing to determine the value of $I_{k,t}$, with smaller ρ favoring $\bar{S}_{k,t}$. The more that $\bar{S}_{k,t}$ affects the measurement, the more that the measurement in turn tells us about $\bar{S}_{k,t}$ and thus the more useful it is to us. When ρ is large, the value of $I_{k,t}$ tends to lie very close to the value of $I_{k,t-1}$ regardless of whether $\bar{S}_{k,t}$ is 1 or -1, and therefore the measurement does not reveal much new information about the syndrome.

1.4 Bayesian Inference and Machine Learning

Before going into the details of our Bayesian inference and machine learning models, let us review the conventional method for error detection in CQEC, namely the double threshold approach.

Double Thresholds

The double threshold protocol from [31] uses two standard signal processing methods, filtering and thresholding, to identify errors. The raw measurement signal is first passed through an exponential filter to smooth out oscillations, and then this averaged value is compared to a pair of adjustable threshold values to determine the state of the system. A slightly different double threshold protocol was proposed in [30], which used boxcar averaging and fixed one of the thresholds at zero.

To estimate the definite error syndromes from the noisy measurements, we first filter the raw signals $I_k(t)$ to obtain corresponding filtered signals $\mathcal{I}_k(t)$ according to

$$\dot{\mathcal{I}}_k(t) = -\frac{\mathcal{I}_k(t)}{\tau} + \frac{I_k(t)}{\tau}, \tag{1.59}$$

where τ is the averaging time parameter, and whose discretized version is similar. In the regime where $t - t_0 \gg \tau$ where t_0 is at the last filtered signal reset, $\mathcal{I}_k(t)$ reads as

$$\mathcal{I}_k(t) = \int_{t_0}^{t} dt' \frac{e^{-\frac{t-t'}{\tau}}}{\tau} I_k(t') . \tag{1.60}$$

After filtering the measurement signals, we then apply a double thresholding protocol to the filtered signals $\mathcal{I}_1(t)$ and $\mathcal{I}_2(t)$ that is parameterized by the two thresholds Θ_1 and Θ_2, where Θ_1 is the threshold for the -1 value of the error syndromes and Θ_2 is the threshold for the $+1$ value of the error syndromes. If at least one of $\mathcal{I}_1(t)$ or $\mathcal{I}_2(t)$ is found to lie within the interval (Θ_1, Θ_2), we declare to be uncertain of the error syndromes and do not perform any error correction operation. Otherwise, we apply the following procedure, in accordance with the standard approach for error diagnosis and correction. If both $\mathcal{I}_1(t) > \Theta_2$ and $\mathcal{I}_2(t) > \Theta_2$, then we diagnose the error syndromes as $(S_1 = +1, S_2 = +1)$ and accordingly perform no error correction operation. If $\mathcal{I}_1(t) < \Theta_1$ and $\mathcal{I}_2(t) > \Theta_2$, then we diagnose the error syndromes as $(S_1 = -1, S_2 = +1)$ and accordingly perform the error correction operation $C_{\text{op}} = X_1$. If both $\mathcal{I}_1(t) < \Theta_1$ and $\mathcal{I}_2(t) < \Theta_1$, then we diagnose the error syndromes as $(S_1 = -1, S_2 = -1)$ and accordingly perform the error correction operation $C_{\text{op}} = X_2$. If $\mathcal{I}_1(t) > \Theta_2$ and $\mathcal{I}_2(t) < \Theta_1$, then we diagnose the error syndromes as $(S_1 = +1, S_2 = -1)$ and accordingly perform the error correction operation $C_{\text{op}} = X_3$.

In quantum annealing, we note that the error correction operations are applied immediately after the error syndromes are diagnosed to minimize the aforementioned spurious Hamiltonian evolution. The action of an error correction operation C_{op}, assumed to be instantaneous, changes the three-qubit state $\rho(t)$ according to

$$\rho(t) \to C_{\text{op}} \rho(t) C_{\text{op}}, \tag{1.61}$$

which applies to other models in our work as well. We note that the parameters $\{\tau, \Theta_1, \Theta_2\}$ constitutes the minimal set of tunable parameters. When the measurement signals I_k have white noise, their optimal values in minimizing the logical error rate can be obtained by Eq. (43) in [31] together with numerical optimizations.

We further reset the filtered signals $\mathcal{I}_k(t)$ to the corresponding initial syndrome value, at the same instant to avoid the transient delay in the filtered signals to reflect the application of the error correction operation on the state. Inherent within any error correction protocol, however, is the implicit assumption that the correction properly removes the error, which may not necessarily be the case if the error was misdiagnosed.

We note that the $\mathcal{I}_k(t)$ used by the double threshold model in CQEC consists of weighted contributions from every raw signal taken prior to t and after the last correction. The discrete

Bayesian model and the RNN-based model that we discuss in this work can both be operated on raw signals, using all historical signals taken prior to a given t. This is in contrast to the projective measurement on ancilla superconducting qubits in discrete QEC that applies a matched filter [47] on raw signals taken only within each detection round.

Discrete Bayesian Classifier

One weakness of the double-threshold scheme is that its predictions are essentially all-or-nothing, since there is no in-built quantity that expresses the model's confidence. This contrasts with probabilistic classifiers, which generate probability values for each prediction class instead of only a single guess. By framing the classification problem in terms of probabilities, we can incorporate our knowledge of the error and noise distributions into our model in a mathematically rigorous manner.

Since each qubit in our system will experience either one or zero net flips after every time step, there are eight different ways that a state can be altered by bit-flips and therefore eight different classes that our classifier must track. We denote each of the possible bit-flip configuration using the state that $|000\rangle$ is taken to by the error, such that $|001\rangle$ denotes a flip on the third qubit, $|110\rangle$ denotes a flip on the first and second qubits, and so on. The goal of a probabilistic error corrector is to accurately determine the probability of all eight "error states" at time step t given the measurement histories $\mathcal{M}_t^k \equiv \{I_{k,t'}\}_{t'=0}^{t'=t}$. We write this posterior probability as

$$\hat{p}(s_t) \equiv p(s_t|\mathcal{M}_t^1\mathcal{M}_t^2), \tag{1.62}$$

where $s_t \in \{0, ..., 7\}$ denotes the digital representation of the error state at step t.

In the remainder of this subsection, we consider a probabilistic classifier constructed using Bayes' theorem, which makes predictions based on the posterior probabilities of the different basis states at each time step [48]. Starting with the knowledge of the initial state, this model uses a Markov chain and a set of Gaussian likelihoods to update our beliefs about the system conditioned on the specific measurement values that we observe.

The Bayesian algorithm described in this section is derived by assuming that the mean of a given measurement $I_{k,t}$ is always determined by the state of the system at the end of the time step. This is equivalent to assuming that errors always happen at the beginning of each time step (see Sec. 1.3). Since our method for generating quantum trajectories follows this assumption, the Bayesian model is theoretically optimal for the numerical tests carried out in Sec. 1.5 without mean drift or resonator transients. As the length of the step Δt between measurements goes to zero, this algorithm converges to the Wonham filter [49], which is known to be optimal for continuous quantum filtering of error syndromes [50]. This filter is similar to the discretized, linear Wonham filter derived in [30], except that our filter does not rely on first-order approximations of the Markov evolution or Gaussian functions.

Using Bayes' theorem, the posterior probability of Eq. (1.62) can be rearranged into the

recursive form

$$\hat{p}(s_t) \propto p(I_{1,t}I_{2,t}|s_t\mathcal{M}^1_{t-1}\mathcal{M}^2_{t-1}) \sum_{i=0}^{7} p(s_t|s_{t-1}=i)\hat{p}(s_{t-1}=i), \tag{1.63}$$

where we assume that the occurrence of an error is independent of any previous measurements and that $I_{k,t}$ depends on the error state at time t along with past signal values due to auto-correlations.

This recursive expression describes a Bayesian filter which takes prior information about the error state of the system and updates it based on the transition probabilities $p(s_t|s_{t-1})$ and measurement likelihoods $p(I_{1,t}I_{2,t}|s_t\mathcal{M}^1_{t-1}\mathcal{M}^2_{t-1})$. The filter can be easily implemented once we have functional forms for these two terms, which we describe next.

The Markovian assumption inherent in $p(s_t|s_{t-1})$ is reasonable, given that the net effect of an additional bit-flip error depends only on the error state the system before the error. We assume hereafter that the error rate γ_q is identical for all three qubits, i.e., $\gamma_q = \gamma$. This allows us to model the errors as a Markov chain [51] with an 8×8 rate matrix Q given by

$$Q_{ij} = \begin{cases} -3\gamma & \text{if } j = i \\ \gamma & \text{if } j \oplus i \in \{1,2,4\} \\ 0 & \text{otherwise,} \end{cases} \tag{1.64}$$

where we define our basis such that index $i \in \{0,\ldots,7\}$ corresponds to the error state whose classical binary representation is equal to i, e.g. $5 \to |101\rangle$.

Since Q only gives the rate of transition per unit time, we need to compute the transition matrix J in order to get probabilities for a finite step. This matrix can be derived from Q as

$$J = e^{Q\Delta t}, \tag{1.65}$$

where Δt is the length of the time step. Element J_{ij} gives the probability of transitioning from error state i to error state j across the time step, so we can relate $p(s_t|s_{t-1})$ to J as $p(s_t = j|s_{t-1} = i) = J_{ij}$. Using J, the sum in Eq. (1.63) can be evaluated to give probabilities $\tilde{p}(s_t)$

$$\tilde{p}(s_t = j) \equiv \sum_{i=0}^{7} \hat{p}(s_{t-1} = i)J_{ij}, \tag{1.66}$$

which take into account the transitions induced by bit-flip errors during the time step.

The measurement likelihood $p(I_{1,t}I_{2,t}|s_t\mathcal{M}^1_{t-1}\mathcal{M}^2_{t-1})$ describes the probability of generating signal values $I_{1,t}$ and $I_{2,t}$ given that the system is in error state s_t and that we had previously measured the values in $\mathcal{M}^1_{t-1}$ and $\mathcal{M}^2_{t-1}$. Since the noise from each syndrome is independent, we can factor the likelihood as

$$p(I_{1,t}I_{2,t}|s_t\mathcal{M}^1_{t-1}\mathcal{M}^2_{t-1}) = p(I_{1,t}|s_t\mathcal{M}^1_{t-1})p(I_{2,t}|s_t\mathcal{M}^2_{t-1}) \tag{1.67}$$

with $I_{1,t}$ and $I_{2,t}$ contributing independently to the probability.

If the noise source is assumed to be Gaussian, then the probability density for each $I_{k,t}$ has the form

$$p(I_{k,t}|s_t\mathcal{M}_{t-1}^k) = \frac{1}{2\pi\sigma^2}\exp\left[\frac{-(I_{k,t}-\mu_{k,t})^2}{2\sigma^2}\right], \tag{1.68}$$

where $\mu_{k,t}$ and σ^2 are the mean and variance of the signal conditioned on the past measurements $\mathcal{M}_{t-1}^k$. In practice the auto-correlations rapidly decay, so we only need to condition on a small number of recent measurements. Hence, we let $m_{k,t-1}$ be the vector of these measurements, and let c be the vector of their corresponding covariance values. Then

$$\mu_{k,t} = \bar{S}_{k,t} + c^T\Sigma^{-1}(m_{k,t-1} - \bar{S}_{k,t}\vec{1}), \tag{1.69}$$

$$\sigma^2 = \frac{\tau}{\Delta t} - c^T\Sigma^{-1}c, \tag{1.70}$$

where $\vec{1}$ is a vector of ones with the same dimension as $m_{k,t-1}$, Σ is the covariance matrix of the variables in $m_{k,t-1}$, and $\bar{S}_{k,t}$ is the mean corresponding to error state s_t [46]. Since the system always begins in the coding subspace, each error state maps to a definite error subspace and therefore has definite syndrome values regardless of how the logical state was initialized.

After the measurement pair $I_{k,t}$ is received, the Gaussian likelihood functions are used to convert the probabilities from Eq. (1.66) into the next posteriors $\hat{p}(s_t)$ as

$$\hat{p}(s_t) \propto \tilde{p}(s_t)\cdot p(I_{1,t}|s_t\mathcal{M}_{t-1}^1)p(I_{2,t}|s_t\mathcal{M}_{t-1}^2), \tag{1.71}$$

which will become probabilities after normalization.

The probabilities from Eq. (1.71) can be understood as describing how likely it is that the system is in each of the eight error states based on the judgment of the model. Whenever $|000\rangle$ does not have the highest probability, we can infer that at least one error has occurred and take the appropriate action to correct it. This procedure, which effectively takes the *argmax* of the posteriors, can be altered if certain forms of misclassification are more costly than others, or if the act of making a correction itself carries some cost. The procedure can also be modified so that it is more robust to imperfections in the signal, as we do in Sec. 1.4 by introducing the τ_{ignore} and τ_{streak} hyperparameters.

Whenever any correction is made, we must update the model with this information by permuting its probabilities to reflect the applied bit-flip. In our example, a correction on the second qubit would lead us to swap the probabilities between pairs of error states which differ in only the second qubit, e.g., $|010\rangle \rightleftharpoons |000\rangle$. Without this update, the model will continue to recommend the same correction repeatedly, as it does not realize that the state of the system has been changed.

A connection can be made between the Bayesian algorithm described here and the maximum likelihood decoder (MLD) commonly used in discrete error correction [52]. Given a specific noise channel and qubit encoding, the MLD is the protocol with the greatest probability of successfully correcting an error, assuming that we have access to *projective* measurements of the syndromes. The Bayesian model can be viewed as an extension of the

MLD to the continuous measurement regime, where the syndrome measurements provide us with incomplete knowledge of the error subspace. As the variance of the Gaussian measurement noise goes to zero, the Bayesian model reduces to the standard MLD protocol for the three-qubit bit-flip code.

Compared to thresholding schemes, the Bayesian classifier described here is far more sensitive to the assumptions we make about the noise and error distributions. Such sensitivity can be an advantage, since it allows for near-optimal performance when our knowledge of these distributions is accurate.

Of course, when our assumptions about the distributions are wrong, the accuracy of the model can suffer significantly. Out of the three imperfections described in Sec. 1.3, only the auto-correlation of neighboring samples is directly accounted for in the model. The resonator transients occur over relatively short time intervals, so they are likely to have only a modest impact on the model's performance. The syndrome drift also has a negative impact, as the mean values of the Gaussian distributions are key parameters in the model. If there is a discrepancy between the actual signal means and our pre-programmed values, then every measurement likelihood calculation will be biased.

We explore the size and significance of these effects for all three of our models in Sec. 1.5.

Recurrent Neural Network

Neural networks are a subset of the broader family of machine learning methods based on acquiring a learned representation of the data, which consists of parameterized layers of linear transformations and nonlinear activation functions. Recurrent neural networks (RNNs) are a class of neural networks in which the layers connect temporally, combining the previous time step and a hidden representation into the representation for the current time step. They are thus well suited for the representation of the time-dependence of continuously measured error syndromes over discrete time steps. Using a training set of labeled signals, the RNN can learn the properties of the weak measurement signal and the structure of the underlying bit-flip channel, which allows it to accurately detect errors as they occur.

The dynamics of a simple recurrent neural network can be expressed by the following equations:

$$\begin{aligned} h_t &= \sigma_h \left(W_h x_t + U_h h_{t-1} + b_h \right), \\ y_t &= \sigma_y \left(W_y h_t + b_y \right). \end{aligned} \tag{1.72}$$

For each time step t, the network accepts the input vector x_t and, along with the hidden state vector from the previous time step h_{t-1}, performs a linear transformation parameterized by the weight matrices W_h and U_h and the bias vector b_h before applying a nonlinear activation function given by σ_h. The result is the hidden state vector for the current time step h_t, which is acted upon by an analogous series of operations defined by W_y, b_y and σ_y to produce the output vector y_t. We note that the hidden state h_t effectively encodes a description of the history of inputs $\{x_{t'}\}_{t'=0}^{t'=t}$, which therefore allows the network to extract temporal, non-Markovian features from the data.

In our context, we consider the input at each time step to be the vector of measurement signals plus the initial basis state,

$$x_t = \begin{bmatrix} I_{1,t} \\ I_{2,t} \\ s_0 \end{bmatrix}. \tag{1.73}$$

Moreover, instead of the standard recurrent neural network architecture, we use a long short-term memory network (LSTM) [53], which is a particular type of recurrent neural network that involves cell states and various gates to evade the vanishing gradient problem of standard RNN architecture [54]. Nevertheless, the same principle underlying the standard function of RNN applies. The output y_t of the LSTM layer is subsequently passed through a dense layer and a softmax activation to produce the posterior probabilities of the eight basis states $p(s_t|\mathcal{M}_t^k)$, and we select the basis state with the highest posterior as the prediction $\hat{s}_t$.

Training samples for the RNN require accurate labeling of the states corresponding to the measurement signals at every time step. However, in reality, decoherence effects such as amplitude damping and thermal excitation prevent us from knowing the correct state of the system at some arbitrary time. As a result, to train the RNN, we have to resort to measurement signals with a well-defined underlying quantum state. This can be achieved by simulating the measurement signals on states in the absence of unwanted decoherence effects, which will be described in detail in Sec 1.5. In the simulations, we provide the measurement strength, the single-qubit bit-flip error rate and the initial quantum state as input parameters, and the simulation produces a large number of quantum trajectories to be the training samples of the RNN. We then train the RNN to diagnose bit-flip errors on the three-qubit system, and the trained RNN can be subsequently used to actively correct errors that occurred. That said, the same information used to generate the training samples is also provided as prior knowledge to the double threshold and the Bayesian model. The two models both require an explicit estimation of the measurement strength as well as the assumption of a certain error rate.

We maximize the likelihood of the RNN parameters on the training set by minimizing the cross-entropy batch total loss function, which is defined as

$$\mathcal{L} = -\frac{1}{NT}\sum_{n=1}^{N}\sum_{t=1}^{T}\log p_n(s_t), \tag{1.74}$$

where $p_n(s_t)$ stands for the posterior probability of the true basis state s_t at time step t in the n-th sample, while N denotes the mini-batch size and T denotes the total number of steps in each training sample.

To update the parameters to minimize the loss, we perform an iterative training procedure where for each step and parameter w, one applies a gradient descent update of the form $w \leftarrow w - \eta(\partial\mathcal{L}/\partial w)$, where the gradients $\partial\mathcal{L}/\partial w$ are computed via backpropagation through the computation graph of the network.

In our experiments, the gradient descent update is performed using the ADAM optimizer [55]. We adopt a two-layer stacked LSTM with a hidden state size of 32. This small

hidden size limits the largest matrix-vector multiplication in computations, hence the memory required, and also limits the number of parameters, facilitating the implementation of the network in real-time experiments. We further provide a comparison test on the performance of different hidden state sizes in Sec. 1.5 and show that both smaller LSTM and gated recurrent unit (GRU) architecture [56] offer comparable performance for our purpose. The number of stacked layers of the LSTM/GRU and the hyperparameters, such as the batch size in training, are tuned with the assistance of Ray Tune [57].

When performing active error correction, we once again wish to avoid the delay in the posterior probabilities output by the network to reflect the application of an error correction operation C_{op} on the system. In the case of the Bayesian classifier, we permute the elements of the vector of posterior probabilities, which encodes the state of the model, in accordance with the error correction operation. For the RNN, however, we cannot apply a particular transformation to the hidden state such that the vector of posterior probabilities outputted by the network is permuted in analogous manner, since the function mapping the hidden state to the output vector of posterior probabilities is highly nontrivial.

Any such delay in the network remaining unaware of the quantum state having been corrected is harmful, because another error X_q occurring during this delay, compounding with the correction C_{op} on the first error, will induce a logical error at the next error correction operation. To see this clearly, considering that the physical qubits are initially in $|000\rangle$, and the first error X_1 results in the state $|100\rangle$. After detecting the error, the model makes a correction that instantly returns the state back to $|000\rangle$. However, the RNN still has the knowledge of the qubits being in $|100\rangle$ until some time later at t_{realize} before accepting a sufficient number of x_t's that allows it to predict $|000\rangle$. If a second error X_2 occurs before t_{realize}, the syndromes become $(S_1 = -1, S_2 = -1)$ because the state becomes $|010\rangle$, whereas the RNN, only knowing the state in $|100\rangle$, will eventually predict $|101\rangle$ that has the same syndromes, which is then equivalent to diagnosing an X_3 error. After applying a second error correction $C_{\text{op}} = X_3$, the physical qubits are now in $|011\rangle$ (since the very first error X_1 has been corrected) and will be corrected to $|111\rangle$ subsequently, constituting a logical error. In other words, since we are not capable of injecting the knowledge of a correction operation into the RNN, a correction operation is equivalent to an error seen by the RNN and active correction effectively increases the bit-flip error rate γ in the eyes of the network. Although the correction is correlated with the detected error, the network is generally trained on quantum trajectories with uncorrelated random bit-flip error instances. As will be explained in 1.5 that a greater γ will induce more logical errors, we conclude that the naive approach of active correction with the RNN suffers from more logical errors.

Therefore, we propose the following re-calibration protocol to effectively hide the action of any error correction operation from the network, so that there is no longer any delay in the posterior probabilities to begin with.

We specifically keep track of all the error correction operations that have been applied up to the present t,

$$N_{q,t} = \text{Number of } X_q \text{ corrections applied.} \tag{1.75}$$

When the measurement signals $I_{1,t}$ and $I_{2,t}$ have symmetric noise around their respective mean values and the possible means of $I_{k,t}$ are always equal and opposite, each C_{op} correction changes the mean of $I_{1,t}$ by a factor of -1 if $C_{\text{op}} = X_1$, changes the mean of $I_{2,t}$ by a factor of -1 if $C_{\text{op}} = X_3$, and changes the mean of both $I_{k,t}$ by a factor of -1 if $C_{\text{op}} = X_2$. To hide all the corrections done in the past, the measurement signals that are provided as input to the network for all subsequent time steps are then flipped according to $N_{q,t}$,

$$\begin{aligned} I'_{1,t} &= (-1)^{N_{1,t}+N_{2,t}} I_{1,t}, \\ I'_{2,t} &= (-1)^{N_{2,t}+N_{3,t}} I_{2,t}, \end{aligned} \tag{1.76}$$

which we called the re-calibrated signals. From the perspective of the RNN when taking in $I'_{k,t}$, it appears as if no error correction operation has been applied to the physical qubits.

Given that at some time step we predict a different state $\hat{s}_t$, we now perform our error correction operation relative to the previous predicted state $\hat{s}_{t-1}$.

When the possible means of $I_{k,t}$ are not equal and opposite, as occurs in the resonator transients upon applying C_{op}, the re-calibration method breaks down, because flipping the means of either or both $I_{k,t}$ does not produce the means as if there was no correction applied. A solution to this is to impose an ignore time period τ_{ignore} right after the correction is applied at some t. During $(t, t + \tau_{\text{ignore}}]$, no input x_t is fed into the RNN. As a result, the hidden state of the network is frozen until the ignore time period ends. The re-calibrated signals are accepted by the network only after $t + \tau_{\text{ignore}}$, which reduces the risk of getting incorrect predictions during the transients, but effectively increases the detection time of any error that occurs during the ignore period.

Imposing τ_{ignore} should be accompanied by a measure to ensure that the RNN diagnoses any error with sufficiently high confidence so that fewer false alarms of error will be followed by an ignore period τ_{ignore} upon correction. A feasible measure in practice is to determine an error correction operation only if the RNN predicts the same state $\{\hat{s}_{t'}\}_{t'=t}^{t'=t+\tau_{\text{streak}}}$ for a streak of time steps τ_{streak} that is different from the old state $\hat{s}_{t-1}$, which is a discrete quantity that is easy to optimize. The $\{\tau_{\text{ignore}}, \tau_{\text{streak}}\}$ then constitutes a minimal set of tunable hyperparameters for the task of active correction in the presence of resonator transients, which applies to the Bayesian classifier explained in Sec. 1.4 as well.

1.5 Simulated Experiments

To evaluate the effectiveness of the three models described in Sec. 1.4, we test their error correction capabilities on a large number of synthetic measurement sequences. The motivation for using artificial data instead of real data is twofold. First, by using artificial data we can precisely control the underlying measurement distribution, which allows us to separate out the effects of the different imperfections identified in Sec. 1.3. Second, it is important that we know the true state of the system at every time step, as this is necessary both to train the RNN and to calculate intermediate fidelity values. Such knowledge would not be possible on a near-term quantum computer due to strong undesirable decoherence.

To ensure that our simulations are grounded in reality, in addition to making idealized simulations for the finite measurement rates without any experimental imperfections, we model them on data taken from a superconducting qubit device. Fig. 1.1 shows measurements taken from this reference data, which consists of approximately 1.6×10^6 sequences lasting 6 µs each[8]. The sequences are comprised of 192 measurement pairs (one for each resonator), sampled every 32 ns. The data contains both "flat" sequences, in which no bit-flip occurs, as well as sequences in which a bit-flip is deliberately applied to one of the three qubits to induce a state transition. Since these bit-flips are all applied at precisely the same time, we are able to track how the signal mean changes during the transient period.

Across all of our tests, we employ four different simulation schemes, each of which is described below. The schemes are designated with letters A–D in order of how much non-ideal behavior they include, with Scheme A having no imperfections and Scheme D having all three imperfections. In all schemes, we ignore the thermal excitation for each qubit, since a typical excitation rate is on the order of $1\,\mathrm{ms}^{-1}$.

Scheme A: Idealized Behavior

In our first scheme, the simulated signal simply conforms to the idealized behavior given by Eq. (1.44). At the beginning of each measurement sequence, the system is set to a specified initial state in the coding subspace, and then the state of the next time step is determined by sampling a number n_q of bit-flips X_q for each qubit from the Poisson distribution, such that $n_q = \exp(-\gamma\Delta t)(\gamma\Delta t)^{n_q}/n_q!$ where Δt is the time step size. These errors are applied to the corresponding qubits to get the next state. This cycle of sampling and propagating errors is repeated until we have generated a sufficiently long sequence of states.

To create the corresponding $I_{k,t}$, we sample a uni-variate Gaussian distribution at each time step with variance $(\Gamma_m^k \Delta t)^{-1}$ and a mean of ±1 determined by the syndrome eigenvalue at that step. Our reference data has

$$\Gamma_m^k \approx 4.7 \times 10^6\ \mathrm{s}^{-1}, \quad \Delta t = 32 \times 10^{-9}\ \mathrm{s}, \quad \eta_k \approx 0.5, \tag{1.77}$$

where Γ_m^k needed to be estimated from the measurement signals while Δt was known to us in advance. This sequence of Gaussian samples plus the underlying states provides a complete description of a system in the context of our error correction task.

Scheme B: Auto-correlations

As a first step away from ideal behavior, we consider noise that is correlated across time. The data generation process for this scheme is effectively the same as that of Scheme A, except that the noise must be sampled sequentially in order to correctly capture the auto-correlations. In our reference data, we find that significant auto-correlations extend back

[8]The 1.6×10^6 sequences break down to about 50,000 sequences for each of the eight initial states and for each of the X_1, X_2, X_3 injected bit-flip or no injected bit-flip.

roughly four steps, with covariance given by

$$c_k^T \approx 5.94 \cdot \begin{bmatrix} 0.61 & 0.25 & 0.1 & 0.05 \end{bmatrix} \tag{1.78}$$

whose ith element is at lag-i. These values were found by taking every contiguous subsequence of length five in our reference data and using them all to compute a covariance matrix. We can simulate Gaussian noise with these auto-correlations one step at a time using Eqs. (1.69, 1.70).

Scheme C: Auto-correlations with Resonator Transients

For our third scheme, we keep the auto-correlations from Scheme B but alter the behavior of the syndrome values so that they include the resonator transients seen in Fig. 1.1 and explained in Sec. 1.3. To incorporate these patterns into our simulation, we first extract the mean values of the transient patterns from our reference data, consisting of 94 steps in total, for each of the twenty-four different single-flip transitions. Our sequence generation process is then identical to Scheme B, except that after an error occurs the next 94 measurements are sampled from Gaussians centered on the transient means instead of the syndrome eigenvalues. The pattern that we use is matched to the state of the system before and after the error. After the transient period has elapsed, the means are set back to ±1 and further samples are generated as usual until another error occurs.

Scheme D: All Imperfections

Our final simulation scheme takes the auto-correlations and resonator transients from Scheme C and adds an underlying drift term to the syndrome means. Since our reference data contains over a million trajectories collected over the span of multiple hours, it is possible to observe significant differences in the syndrome means between trajectories that are separated by large amounts of time, possibly due to temperature fluctuations.

For our experiments, we elected to apply a linear drift Δ_i governed by

$$\Delta_i = \frac{0.4}{N} \cdot i, \tag{1.79}$$

where i is an index that arbitrarily orders the different measurement sequences that we generate and N is the total number of these sequences. This drift term is added to every measurement in the ith sequence, resulting in a uniform shift of the overall signal means. The net drift across all runs represents a 40% change, which is consistent with the magnitude of the drift observed in our reference data.

Quantum Memory State Tracking

In quantum memory, it suffices to track the basis states in response to the bit-flip errors that have occurred and only apply error correction operations when needed. We generated

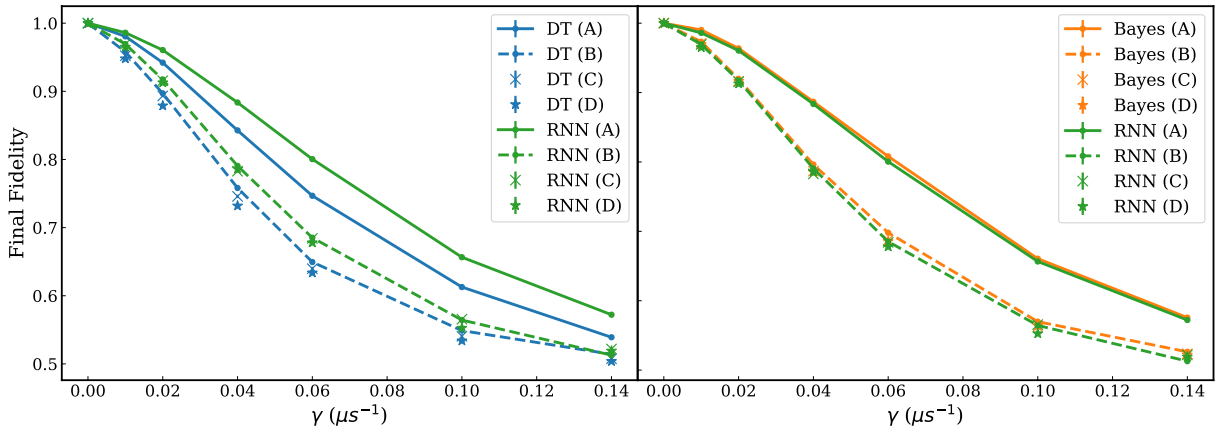


Figure 1.4: The final fidelity with respect to the initial state $|000\rangle$ in Schemes A, B, C, D with the double threshold (DT), Bayesian and RNN classifier, as a function of single-qubit bit-flip rate γ at an operation time $T = 20\,\mu s$. Each data point is averaged over $30,000$ quantum trajectories. For better visualization, we split the figure into two plots, with the left one comparing the RNN classifier to the double threshold, and the right one comparing the RNN classifier to the Bayesian classifier. On the left, we see that the RNN classifier outperforms the double threshold in all schemes. Whereas on the right, it shows that the RNN approximates the Bayesian classifier, which is the optimal one among the three, in all schemes. The error bars show the standard error of the mean.

$30,000$ trajectories of length $T = 20\,\mu s$ from all four simulation schemes with a pre-defined single-qubit error rate as our testing samples, among which are equal portions of trajectories initialized in one of the eight basis states. While the RNN model employed here is trained on $100,000$ quantum trajectories from the corresponding simulation scheme, the error rate, noise variance, and auto-correlations input to the Bayesian model are also estimated from those quantum trajectories. The tunable parameters in the double threshold model are numerically optimized in schemes with imperfections; the filtering time τ typically lies in the range $0.3 - 1.6\,\mu s$, with larger τ for smaller γ.

In Fig. 1.4, we compare the final fidelity $\mathcal{F} = |\langle\psi_T|\psi_0\rangle|^2$ against the initial state of the three models in tracking these quantum trajectories subject to bit-flips. The trend is that the final fidelity decreases as a function of the single-qubit error rate γ. This is because the higher the error rate is, the more chances there will be two different bit-flips before the correction to the first bit-flip is made, resulting in a logical error upon the correction, and therefore a lower final fidelity. For instance, a state starting at $|000\rangle$ is flipped to $|001\rangle$ at t_1 and is later also flipped to $|011\rangle$ at $t_2 > t_1$, such that t_2 is smaller than $t_1 + t_{\text{detect}}$ where t_{detect} is the detection time of the first error. Subsequently, the model perceiving syndromes with $(S_1 = -1, S_2 = +1)$ will eventually make a $C_{\text{op}} = X_1$ correction and change the state to $|111\rangle$, leading to a logical error. From the above argument, it is also evident that a shorter

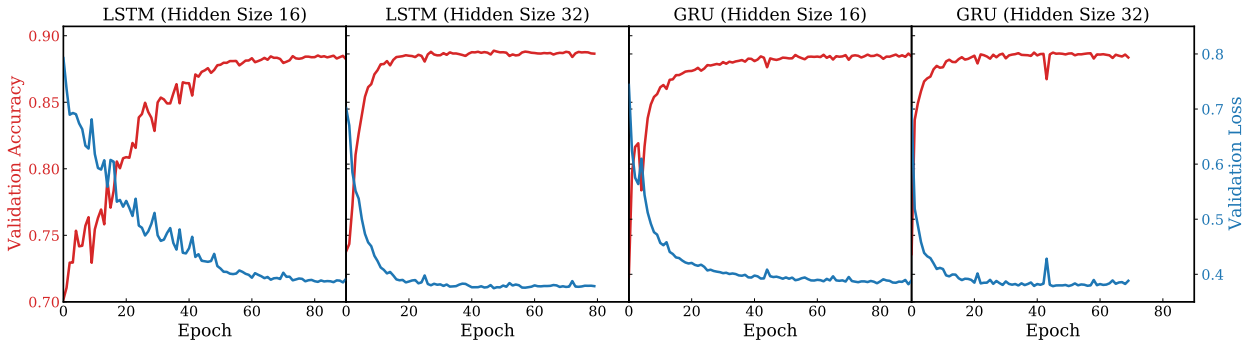


Figure 1.5: The learning curves of LSTMs with hidden sizes of 16 and 32, and of GRUs with hidden sizes of 16 and 32, on the state tracking task in quantum memory as described in Sec. 1.5. The accuracy is defined to be the fidelity with respect to the initial state averaged across all time steps, and the loss is computed by Eq. (1.74).

detection time is beneficial.

From Fig. 1.4, we see that the RNN and the Bayesian classifier outperform the double threshold in all simulation schemes, whereas the RNN approximates the Bayesian classifier in all schemes. As discussed in Sec. 1.4, the Bayesian classifier is the optimal model of the three in Schemes A and B where there are only auto-correlations in the signals, which is validated in this task. The fact that their performances in Schemes C and D are very similar to that in Scheme B indicates that the resonator transient pattern and the drifting of the means do not have a significant effect on all three models.

It is reasonable that the drift has a small negative effect to the two probabilistic models, since the drift is usually on the order of the separation of mean values of the two parities, which is in turn one order of magnitude smaller than the standard deviation of the noise. The large noise variance obscures the drifting means, making the drifted signals appear like more noisy signals with fixed means.

RNN Hidden State Size v.s. Performance

It is desirable to limit the size of the RNN to achieve sufficiently low computational latency in real-time experiments. We present the performance in state tracking in quantum memory as described in Sec. 1.5 for the LSTM and GRU architectures with different hidden sizes in Tab. 1.1. In examining the performance, we see that although we used LSTM with a hidden size 32 in our simulated experiments, it is possible to shrink the size of the network to 16 without harming the performance. We note that a smaller hidden size means smaller matrix-vector multiplications in computing the model, which then requires fewer memory resources in practice. The possible simplification is also suggested by the fact that the learning curves with a hidden size of 16 is very similar to that with a hidden size of 32, as shown in Fig. 1.5. Additionally, it is viable to use the GRU architecture to achieve the same performance.

Table 1.1: The testing performance of LSTM (top) and GRU (bottom) with different hidden sizes and the corresponding number of trainable parameters. The testing performance is measured by the final excited states population P_{exc}. The hidden size determines the largest matrix-vector multiplication operation performed when computing the model.

Hidden size	8	16	32	64
Parameter count	1064	3256	13448	51464
Final P_{exc} (± 0.002)	0.851	0.880	0.884	0.882
Hidden size	8	16	32	64
Parameter count	816	2776	10152	38728
Final P_{exc} (± 0.002)	0.816	0.880	0.879	0.881

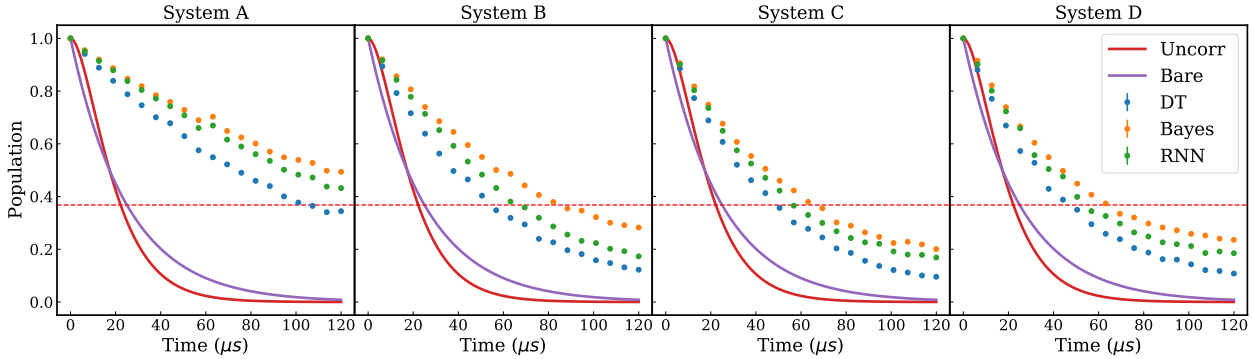


Figure 1.6: The population of the excited states $\{|111\rangle, |110\rangle, |101\rangle, |011\rangle\}$ as a function of time, obtained from simulated experiments with the four different schemes at a single-qubit decay rate of $\gamma = 0.04\,\mu\text{s}^{-1}$. Each data point is averaged over 3,000 independent quantum trajectories. The three-qubit system is initialized to $|1\rangle_{\text{L}} = |111\rangle$. As a comparison, the bare qubit (purple curve) is initialized to the $|1\rangle$ state and is subject to amplitude damping with a time constant of $T_1 = 25\,\mu\text{s}$, i.e., a decay rate of $0.04\,\mu\text{s}^{-1}$. For reference, the uncorrected three-qubit system decay curve is shown in red (see Sec. 1.5). For all schemes, the RNN-based model outperforms the double threshold model.

These results suggest that the RNN-based model may have a simpler structure and an even faster computation speed in real-time implementation on programmables like FPGAs.

We note that the size of the RNN can be further reduced, if assuming a fixed initial state so that the input to the RNN shown in Eq. (1.73) can be replaced by $x = [I_{1,t}, I_{2,t}]^T$.

Extending T_1 Time of the Logical Qubit

Although the models are motivated by correcting bit-flip errors, they can also be exploited in extending the T_1 time of the logical qubit in $|1\rangle_{\text{L}} = |111\rangle$. For this task, actively correcting

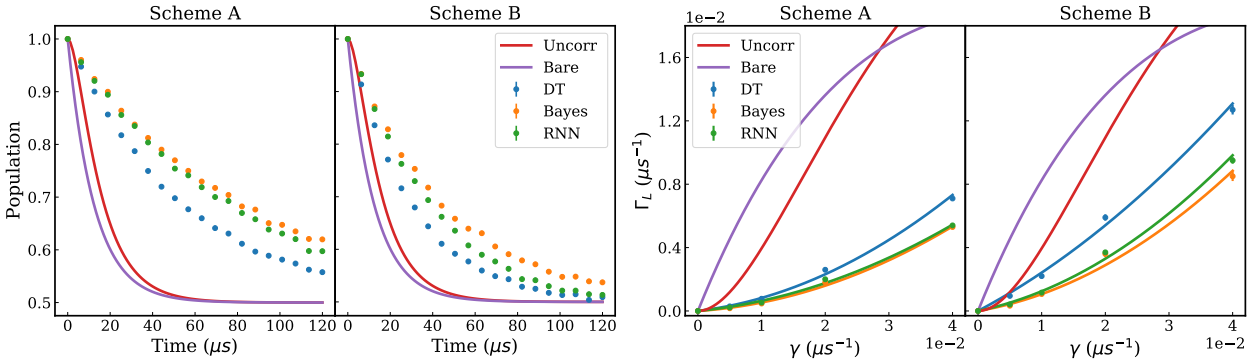


Figure 1.7: Left: the population of the excited states $\{|111\rangle, |110\rangle, |101\rangle, |011\rangle\}$ as a function of time, obtained from simulated experiments under Schemes A and B at a single-qubit bit-flip rate of $0.04\,\mu s^{-1}$. Each data point is averaged over $3,000$ independent quantum trajectories. The three-qubit system is initialized to $|1\rangle_L = |111\rangle$. As a comparison, the bare qubit (purple curve) is initialized to $|1\rangle$ and is subject to a bit-flip rate of $\gamma = 0.04\,\mu s^{-1}$. As a reference, the uncorrected three-qubit system decay curve is shown in red (see Sec. 1.5). In Schemes A and B, the Bayesian model is the best among the three, and the Bayesian and RNN-based models both outrun the double threshold model. Right: the initial logical error rate Γ_L at $9.6\,\mu s$ as a function of the single-qubit error rate γ. The fitted quadratic curves show a strong suppression of Γ_L for all three models in both schemes.

the state is required as opposed to merely tracking the state. While for practical purposes the RNN model is trained on $30,000$ quantum trajectories under bit-flips with a length of $T = 120\,\mu s$, the Bayesian model, whose parameters are estimated from the same set of trajectories, uses a different transition matrix generated by Q' shown in Eq. (1.80) which takes into account the asymmetric probabilities of transitions between the ground and excited state. The parameters for the double threshold model is numerically optimized on the same set of quantum trajectories.

For the three-qubit system initialized to the fully excited state $|111\rangle$, we inspect the population within a Hamming distance 1 away from the initial state, i.e., the population P_{exc} of the set of basis states $\{|111\rangle, |110\rangle, |101\rangle, |011\rangle\}$, since these states can be recovered to the initial state by a majority vote. We compare this P_{exc} against the population of the excited state $|1\rangle$ of a bare qubit as a function of time in all four simulation schemes, and the results are shown in Fig. 1.6. In all schemes, the encoded three-qubit system P_{exc} decays much slower under active correction by any of the three models than the bare-qubit excited state population. In all schemes, both the Bayesian and the RNN-based models outrun the double threshold model.

Protecting against Bit-flip Errors

Similar to the task of extending the T_1 time of the state $|1\rangle_L$, here we employ the three models to protect the initial state $|1\rangle_L$ from bit-flips. Shown in Fig. 1.7, we compare the population P_{exc} of the three-qubit system against the excited population of the bare qubit in time. For Schemes A and B, both the Bayesian and the RNN-based models have an advantage over the double threshold. Furthermore, in Fig. 1.7 we extract the initial logical error rate Γ_L as a function of γ by computing the time derivative of P_{exc} at 9.6 µs at each γ. In either scheme with any of the three models, Γ_L scales approximately quadratic in γ, and we can see a strong suppression of Γ_L relative to a bare qubit or the uncorrected three qubits. We remark that, by introducing feedback based on noisy weak measurements, any correction protocol can underperform a majority vote on the encoded qubits without error correction at sufficiently small γ or runtime.

To better understand the performance of the models in this important task, we analyze the detection time spent in true positive detection as well as the number of false alarms when the three-qubit system is in $|1\rangle_L$. The difference between a true positive and a false alarm is illustrated in Fig. 1.8, which shows the actual and predicted states of the system when an X_3 error occurs and when the model falsely detects an X_1 error. When a true error occurs, the system remains in the corresponding error subspace for a duration determined by the detection time of the model, after which the error is corrected. By contrast, when the model falsely detects that an error has occurred due to measurement noise, it improperly applies a bit-flip to the system and thus pushes it out of the code subspace. After more measurements are recorded, the model determines that the system is in an error subspace and fixes its mistake by applying another bit-flip.

As explained in Sec. 1.5, a shorter detection is favorable and will lead to better error corrections, whereas here we can expect more frequent false alarms arise for models with a shorter detection time as a trade off, since the model is prone to make a correction. This is demonstrated in Fig. 1.9, where we can see that the best two models, the Bayesian and the RNN-based, both have a shorter detection time and more frequent false alarms at the same time. Nevertheless, for both of these two models, the overall frequency of all false positive detection remains low and is on the order of 0.1 μs^{-1}.

Population of States Subject to Amplitude Damping or Bit-flips

We recall that the population of the excited states P_{exc} is the ensemble population of the states that are at most one bit-flip away from the fully excited state $|111\rangle$, i.e., $P_{exc} = P(|111\rangle) + P(|110\rangle) + P(|101\rangle) + P(|011\rangle) = P_7 + P_6 + P_5 + P_3$.

Under T_1 decay at zero temperature, the transition matrix evolving the states for time

T is $J'(T) = \exp(Q'T)$, where Q' is defined as,

$$Q' = \gamma \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & -2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & -2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & -2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & -3 \end{bmatrix}. \tag{1.80}$$

The state probabilities under the Markov chain are given by $P(T) = J'(T)P(0)$, which yields

$$P_{\text{exc}}(T) = \left(3e^{\gamma T} - 2\right) e^{-3\gamma T}. \tag{1.81}$$

Under only bit-flip errors X_q, the transition matrix evolving the states for time T is $J(T) = \exp(QT)$, where Q is defined in Eq. (1.64). The resultant population of excited states is

$$P_{\text{exc}}(T) = e^{-3\gamma T} \cosh^2(\gamma T) \left[3 \sinh(\gamma T) + \cosh(\gamma T)\right]. \tag{1.82}$$

This can be equivalently derived from the probability of a qubit experiencing an even/odd number of flips under Poisson-distributed bit-flip errors. The probability that the k-th qubit will experience an even number of bit-flips is

$$P(e_k \text{ is even}) = \sum_{j=0}^{\infty} P(e_k = 2j) = e^{-\gamma T} \sum_{j=0}^{\infty} \frac{(\gamma T)^{2j}}{(2j)!} = e^{-\gamma T} \cosh(\gamma T). \tag{1.83}$$

and thus the probability that the k-th qubit will experience an odd number of bit-flips is

$$P(e_k \text{ is odd}) = 1 - P(e_k \text{ is even}) = e^{-\gamma T} \sinh(\gamma T). \tag{1.84}$$

Using the above two equations, we recover Eq. (1.82).

Quantum Annealing with Time-dependent Hamiltonians

Having demonstrated a clear advantage using the RNN-based protocol for tasks in the quantum memory setting over the double threshold protocol, we now study the performance of our protocol for quantum annealing, using a time-dependent Hamiltonian that does not non-commute with the bit-flip errors. We note that the protocol is also applicable to evolution under quantum gate operations.

In quantum annealing, it is imperative to perform error diagnosis and correction in a manner that is both fast and accurate, in order to avoid accruing these logical errors while single bit-flip errors are being diagnosed and corrected. This is because the action of an error X_q effectively transforms the Hamiltonian from $H(t)$ to $X_q H(t) X_q$ in the Heisenberg

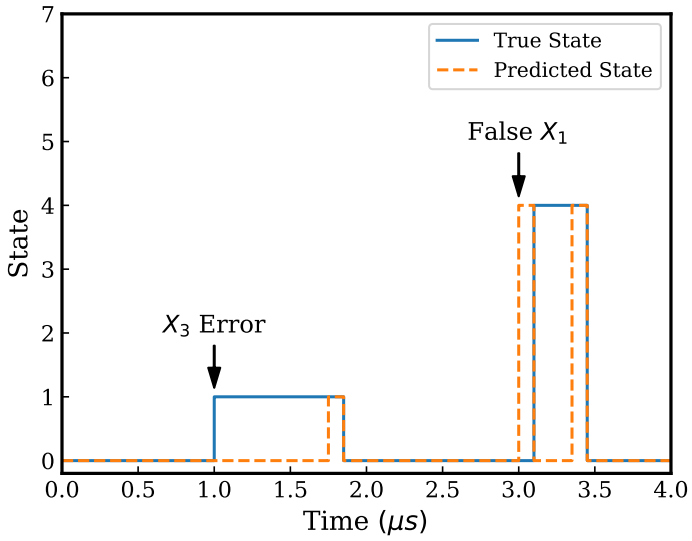


Figure 1.8: Response of the system basis state and model to a true bit-flip error and a false alarm as a function of time. At 1.0 µs an X_3 error is applied to the system, and after a small delay the error is detected and corrected. At 3.0 µs the model falsely detects and then "corrects" for an X_1 error, which results in the system being temporarily pushed into an error subspace before the mistake is recognized and corrected. There are visible small constant offsets between the prediction and the system state at the false alarm due to the streak time period imposed in the correction protocol.

picture. Until the error is properly diagnosed and corrected, subsequent coherent evolution of the logical state in the code subspace is due to the modified Hamiltonian $X_q H(t) X_q$. If the original Hamiltonian does not commute with the error, i.e. $X_q H(t) X_q \neq H(t)$, then such evolution will be spurious rather than as originally intended, causing logical errors to accrue.

We adopt the jump/no-jump method for bit-flip errors. In this method, gradual decoherence due to the third term in Eq. (1.46) is described as the average effect of bit-flip errors X_q occurring at random times. At a finite time interval $[t, t + \Delta t]$, a bit-flip error X_q occurs with probability $\gamma_q \Delta t$. If this error occurs, the quantum state jumps from ρ_t to $\rho_{t+\Delta t} = X_q \rho_t X_q$. Otherwise, the quantum state continuously evolves without environmental decoherence. Upon averaging over many instances of the bit-flip errors, the jump/no-jump approach reduces to the open quantum system model, where errors continuously change the mixed system state $\rho(t)$.

In simulating the coherent evolution, we use the first-order Magnus expansion [58] of the annealing Hamiltonian $H(t)$ in Eq. (1.85) at every finite time interval $[t, t + \Delta t]$, $\tilde{U}_t = \exp[-iH(t')\Delta t]$ where $t' = t + \Delta t/2$, such that the quantum state evolves as $\rho_{t+\Delta t} = \tilde{U}_t \rho_t \tilde{U}_t^\dagger$.

We average over 10,000 quantum trajectories obtained through the above-mentioned steps to simulate the ensemble density states ρ_t.

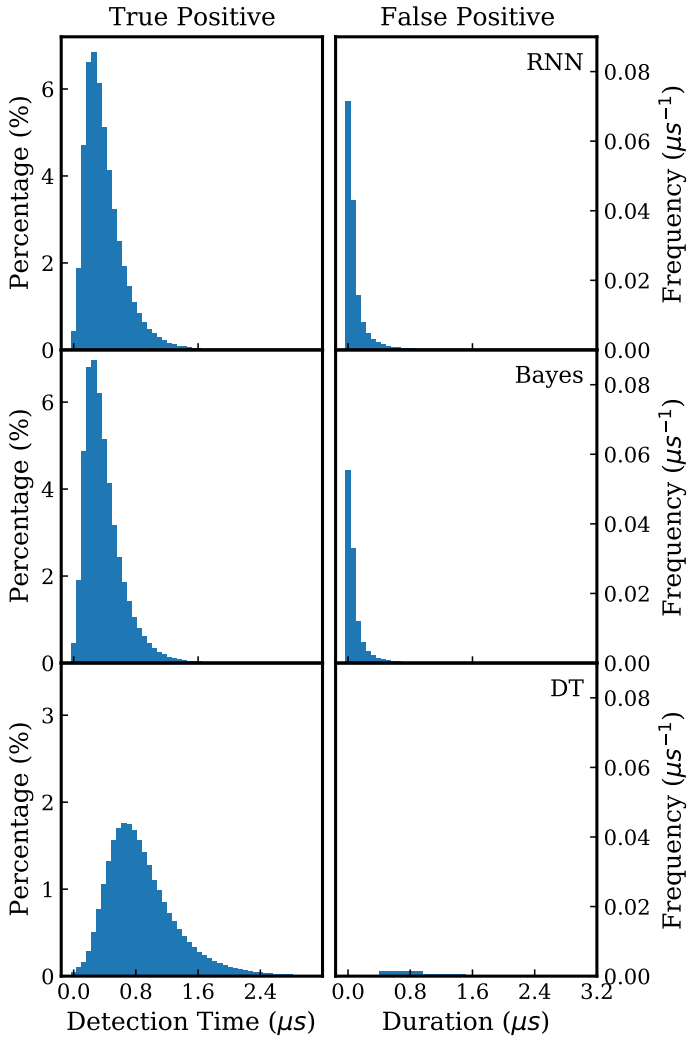


Figure 1.9: The distribution of detection time (with the left y-axis) and the distribution of false alarms of bit-flips (with the right y-axis) when the state is originally in $|111\rangle$, over $100,000$ quantum trajectories with an operation time $T = 120\,\mu s$ and with a single-qubit bit-flip rate $\gamma = 0.04\,\mu s^{-1}$. The three qubits are initialized to $|111\rangle$. The overall frequencies of all false alarms for the RNN-based, Bayesian, and double threshold models are 0.155(5), 0.117(2), 0.0022(2) μs^{-1}, respectively.

For this simulated experiment, the annealing Hamiltonian with a strength Ω_0 evolving $\rho_0 = |\psi_0\rangle\langle\psi_0|$, $|\psi_0\rangle = (|0\rangle_L + |1\rangle_L)/\sqrt{2}$ is chosen to be

$$H(t) = -\Omega_0 \left[a(t) X_1 X_2 X_3 + b(t) \frac{Z_1 + Z_2 + Z_3}{3} \right], \tag{1.85}$$

where $a(t) = 1 - t/T$ and $b(t) = t/T$. In the code subspace, it is equal to

$$h(t) = -\Omega_0 \left[a(t)\sigma_x + b(t)\sigma_z \right], \tag{1.86}$$

whereas in any error subspace, it is equal to the spurious Hamiltonian,

$$h_{\text{spurious}}(t) = -\Omega_0 \left[a(t)\sigma_x + b(t)\frac{\sigma_z}{3} \right]. \tag{1.87}$$

We adopt the reduction factor [31] as the metric for evaluating the model performance, which is defined as,

$$\mathcal{R} = \frac{1 - \mathcal{F}_{\text{une}}}{1 - \mathcal{F}}, \tag{1.88}$$

whose numerator is the final infidelity of an unencoded bare qubit initialized to $|0\rangle$ under the annealing Hamiltonian Eq. (1.86), and whose denominator is the final infidelity of the three-qubit encoded state in the code subspace with respect to the target quantum state. As $\dot{a}(t), \dot{b}(t) \to 0$, the target quantum state becomes the ground state of the target Hamiltonian.

As shown in Fig. 1.10, at relatively low γ, the Bayesian model achieves the highest reduction factor in Scheme A, while both the Bayesian and the RNN-based model outperform the double threshold. However, at sufficiently high error rates γ, the encoded qubits under active correction using any of the three models show no improvement over a single unencoded qubit, as expected.

1.6 Discussion

We have proposed an RNN-based CQEC algorithm that is able to outperform the popular double threshold algorithm across all tasks for each of the four simulation schemes tested in Sec. 1.5. This result holds regardless of whether the algorithms are protecting a system from bit-flip errors or from amplitude damping, and applies in the case of both quantum memory and quantum annealing. The relative performance of the three models does not depend significantly on the underlying error rate or the duration of the experiment, unless either of these values is exceptionally large.

The mathematical simplicity of Eq. (1.44) is a product of many idealized assumptions, so we can expect that measurements taken from real quantum devices will not necessarily be as easy to describe. Our analysis of superconducting qubit measurements in Sec. 1.3 reveals several examples of non-ideal behavior in both the syndrome and noise distributions, and we expect similar findings in the outputs of other devices. While some signal imperfections can be accounted for in traditional CQEC algorithms, such as the incorporation of

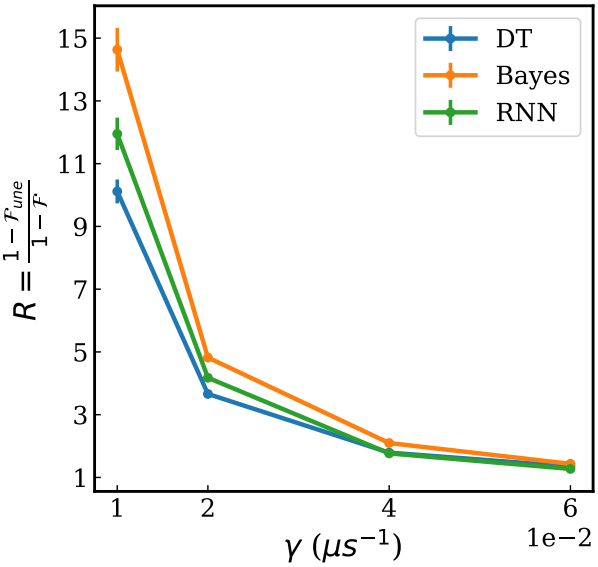


Figure 1.10: The final infidelity reduction factor as a function of single-qubit bit-flip rate γ, with an operation time $T = 120\,\mu s$, and the strength of the annealing Hamiltonian in Eq. (1.85) equal to $\Omega_0 = 0.04\Gamma_m$ where the measurement strength is set to $\Gamma_m = 4.7\,\mu s^{-1}$. The quantum efficiency is set to $\eta = 0.5$. Each data point is averaged over $10,000$ quantum trajectories.

auto-correlations into the Bayesian classifier, most of them will not be easy to precisely characterize. It is in these situations that neural networks can best demonstrate their advantage, since they do not require any a priori description of the patterns within the measurement signals, but instead learn them directly from the training data. An interesting direction for further study is the extension of the RNN-based CQEC algorithm to correlated and leakage errors.

A CQEC algorithm should be practical to run on a sub-microsecond timescale, typically using an FPGA or other programmable, low-latency device. The Bayesian model requires division to normalize the posteriors, which is a very costly operation on FPGAs. This makes it challenging to efficiently implement the Bayesian model, although a more practical log-Bayesian approach has recently been developed [59]. The RNN-based model, by contrast, does not require division and avoids this problem. There are many precedents for running RNNs on FPGAs (see e.g. [60]). Since the RNN architecture used in our study is small in size (more simplifications are discussed in Sec. 1.5), its computational latency is sub-microsecond. Nevertheless, more work will be needed in order to determine how best to interface the RNN with the quantum computer in a feedback loop. For supervised learning, there is the need for generating a sufficient amount of training data that incorporates the error information and the signal features. Further work could focus on determining the minimum amount and

type of data that the RNN needs to train effectively, and understand how these needs change as the number of physical qubits in the error code increases.

Given low-latency implementations of the Bayesian and RNN-based models, an obvious next step for future work would be a direct comparison between these CQEC protocols and existing discrete QEC protocols on quantum hardware. Ristè *et al.* [61] have already demonstrated discrete QEC for a three-qubit bit-flip code on transmons, and recent work by Livingston *et al.* [32] has implemented a triple threshold CQEC protocol on similar hardware. By running experiments on a given physical device, a full comparison between discrete and continuous CQEC can be made under realistic conditions. Due to the lack of both entangling gates and ancillas, we are optimistic that CQEC could significantly improve the speed and fidelity of many QEC codes.

Chapter 2

Practical Quantum Error Mitigation

This chapter is derived from work by Liao, Wang, Sitdikov, Salcedo, Seif, and Minev [40], where Liao and Wang are the co-first authors, which proposed a machine learning framework for efficiently and effectively mitigating observable expectation values of both small- and large-scale quantum circuits.

2.1 Background on Quantum Error Mitigation

Quantum computers promise remarkable advantages over their classical counterparts, offering solutions to certain key problems with speedups ranging from polynomial to exponential [62, 63, 64]. Despite significant progress in the field, the practical realization of this advantage is hindered by inevitable errors in the physical quantum devices. In principle, reduced error rates and increased qubit numbers will eventually enable fully fault-tolerant quantum error correction to overcome these errors [65]. While this goal remains far out of reach at scale, quantum error mitigation (QEM) strategies have been developed to harness imperfect quantum computers to nonetheless yield near noise-free and meaningful results despite the presence of unmonitored errors [66, 5, 67, 68, 69, 70]. QEM is paving the way to near-term quantum utility and a path to outperform classical supercomputers [63, 70].

The main challenge to employing QEM in practice is devising schemes that yield accurate results without excessive runtime overheads. For context, quantum error correction relies on overheads in qubit counts and real-time monitoring of errors to eliminate errors for each run of a circuit. In contrast, QEM obviates the need for both of these overheads but at the cost of increased algorithmic runtime. QEM instead yields an estimator for the noise-free expectation values of a target circuit, the results of a computation, by employing an ensemble of many noisy quantum circuits. For example, in the cornerstone QEM approach known as zero-noise extrapolation (ZNE) [71, 72], an input circuit is recompiled into multiple circuits that are logically equivalent but each with an expected increased total number of errors. From the dependence of the measured expectation values for each noisy circuit, one can estimate the ‘zero-noise’, ideal expectation value of the original circuit. While ZNE does not

yield an unbiased estimator, other QEM methods, such as probabilistic error cancellation (PEC) [71, 72, 68] do and come fortified with rigorous theoretical guarantees and sampling complexity bounds. Unfortunately, it is believed that QEM methods demand exponential sampling overheads [73, 74]—the intuition comes from that it requires an exponentially large sampling overhead for the error bar to be small enough to resolve the exponential decay of the diagonals resulted from depolarizing noise as shown in Eq. (1.5) in Chapter 1—making them challenging to implement at increasing scales of interest.

The quest for QEM methods that strike a balance between scalability, cost-effectiveness, and generality remains at the forefront of quantum computing research. We shall briefly review some important existing QEM methods, before introducing our new approach based on machine learning.

Randomized Compiling

Randomized compiling [68, 75] involves the strategic introduction of randomness into the sequence of quantum gates within a quantum circuit to mitigate errors. This is primarily achieved through a technique called Pauli twirling. In this method, each Clifford gate[1] in the circuit is sandwiched between pairs of randomly selected twirling gates $T \in \{I, X, Y, Z\}$ and their inverses such that the circuit remains logically unchanged. Suppose we have some easy (usually single-qubit), noisy gates $\tilde{\mathcal{C}}$, as well as some hard (usually entangling), noisy gates $\tilde{\mathcal{G}}$ in a circuit, and they are separated into different cycles, each of which consists of (successive) easy noisy gates followed by a hard noisy gate, for instance, the k-th cycle is $\rho_k = (\tilde{\mathcal{G}} \circ \tilde{\mathcal{C}})(\rho_{k-1}) = G\Lambda_{\tilde{\mathcal{G}}}(\Lambda_{\tilde{\mathcal{C}}}(C\rho_{k-1}C^\dagger))G^\dagger$. Twirling a cycle of the noisy circuit is to perform the following conjugation of the hard noisy gate $G = T^c G T$ such that

$$\begin{aligned} GC\rho_{k-1}C^\dagger G^\dagger &= T^c G T C \rho_{k-1} C^\dagger T^\dagger G^\dagger T^{c\dagger} \\ G\Lambda_{\tilde{\mathcal{G}}}(\Lambda_{\tilde{\mathcal{C}}}(C\rho_{k-1}C^\dagger))G^\dagger &\mapsto T^c G \Lambda_{\tilde{\mathcal{G}}}(\Lambda_{\tilde{\mathcal{C}}}(TC\rho_{k-1}C^\dagger T^\dagger))G^\dagger T^{c\dagger} \end{aligned} \tag{2.1}$$

where $T^c = GT^\dagger G^\dagger$. These additional gates create a "twirl" effect, effectively randomizing any coherent errors in $\Lambda_{\tilde{\mathcal{G}}} \circ \Lambda_{\tilde{\mathcal{C}}}$, transforming them into stochastic Pauli errors[2]. The original computational action of the circuit remains unchanged. Any expectation value is then averaged over all these randomly sampled (sampling the T) twirling circuits. These stochastic Pauli errors are less harmful than the original coherent errors in terms of the effect on the expectation values, thus we effectively mitigate the noisy expectation values simply by randomized compiling. We remark that single-qubit twirling gates such as those used in Pauli twirling can be combined with the single-qubit easy gates in the circuit, reducing the additional overhead incurred by twirling in practice.

To see why randomized compiling is able to covert coherent errors into incoherent ones, it is enlightening to look at the Pauli transfer matrix (PTM), $\mathrm{PTM}_{ij}(\mathcal{E}) = \mathrm{Tr}[P_j \mathcal{E}(P_i)]$ of a

[1] Non-Clifford gates can be partially twirled by sandwiching Pauli gates that remain Pauli upon commutation with the non-Clifford gate.

[2] An n-qubit Pauli channel Λ is a quantum channel of the following form $\Lambda(\cdot) = \sum_{a \in \mathbb{Z}_n^{2n}} p_a P_a(\cdot) P_a$.

single-qubit error channel $\Lambda(\cdot)$,

$$\mathrm{PTM}_{ij}(\Lambda) = \mathrm{PTM}_{ij}(I\Lambda(\cdot)I) = \begin{array}{c} \\ I \\ X \\ Y \\ Z \end{array} \begin{array}{c} \begin{array}{cccc} I & X & Y & Z \end{array} \\ \begin{pmatrix} f_{II} & f_{IX} & f_{IY} & f_{IZ} \\ f_{XI} & f_{XX} & f_{XY} & f_{XZ} \\ f_{YI} & f_{YX} & f_{YY} & f_{YZ} \\ f_{ZI} & f_{ZX} & f_{ZY} & f_{ZZ} \end{pmatrix} \end{array}. \tag{2.2}$$

It can be readily shown that the PTM is diagonal if and only if the error channel is a stochastic Pauli channel (a type of incoherent noise). Next, let us look at the Pauli-twirled version of the channel $P\mathcal{E}(\cdot)P^\dagger$,

$$\mathrm{PTM}_{ij}(X\Lambda(\cdot)X) = \begin{array}{c} \\ I \\ X \\ Y \\ Z \end{array} \begin{array}{c} \begin{array}{cccc} I & X & Y & Z \end{array} \\ \begin{pmatrix} f_{II} & f_{IX} & -f_{IY} & -f_{IZ} \\ f_{XI} & f_{XX} & -f_{XY} & -f_{XZ} \\ -f_{YI} & -f_{YX} & f_{YY} & f_{YZ} \\ -f_{ZI} & -f_{ZX} & f_{ZY} & f_{ZZ} \end{pmatrix} \end{array},$$

$$\mathrm{PTM}_{ij}(Y\Lambda(\cdot)Y) = \begin{array}{c} \\ I \\ X \\ Y \\ Z \end{array} \begin{array}{c} \begin{array}{cccc} I & X & Y & Z \end{array} \\ \begin{pmatrix} f_{II} & -f_{IX} & f_{IY} & -f_{IZ} \\ -f_{XI} & f_{XX} & -f_{XY} & f_{XZ} \\ f_{YI} & -f_{YX} & f_{YY} & -f_{YZ} \\ -f_{ZI} & f_{ZX} & -f_{ZY} & f_{ZZ} \end{pmatrix} \end{array}, \tag{2.3}$$

$$\mathrm{PTM}_{ij}(Z\Lambda(\cdot)Z) = \begin{array}{c} \\ I \\ X \\ Y \\ Z \end{array} \begin{array}{c} \begin{array}{cccc} I & X & Y & Z \end{array} \\ \begin{pmatrix} f_{II} & -f_{IX} & -f_{IY} & f_{IZ} \\ -f_{XI} & f_{XX} & f_{XY} & -f_{XZ} \\ -f_{YI} & f_{YX} & f_{YY} & -f_{YZ} \\ f_{ZI} & -f_{ZX} & -f_{ZY} & f_{ZZ} \end{pmatrix} \end{array}.$$

It can be readily seen that an average over the Pauli conjugation yields an error channel that is a stochastic Pauli channel, namely, the PTM of $\frac{1}{4}\sum_{P\in\{I,X,Y,Z\}} P\Lambda(\cdot)P$ is diagonal.

It remains to show that for the same error parameter magnitude, a stochastic Pauli channel is less harmful than a coherent error channel in terms of the expectation value. It is helpful to look at the single-qubit case, with the coherent error being an over-rotation of angle $m\theta$ (m over-rotations of θ combined), $R_X(m\theta)$,

$$\mathrm{PTM}_{ij}(R_X(m\theta)) = \begin{array}{c} \\ I \\ X \\ Y \\ Z \end{array} \begin{array}{c} \begin{array}{cccc} I & X & Y & Z \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos(m\theta) & -\sin(m\theta) \\ 0 & 0 & -\sin(m\theta) & \cos(m\theta) \end{pmatrix} \end{array}. \tag{2.4}$$

If we twirl every $R_X(\theta)$, the PTM of m over-rotations combined, $(PR_X(\theta)P)^m$, is

$$\mathrm{PTM}_{ij}((PR_X(\theta)P)^m) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & \cos(\theta) \end{pmatrix}^m = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos^m(\theta) & 0 \\ 0 & 0 & 0 & \cos^m(\theta) \end{pmatrix}. \quad (2.5)$$

Measuring in the Z basis amounts to looking at the last entry of the PTM—from Eq. (2.4), we have for the coherent error $\langle Z\rangle_{\text{noisy}} - \langle Z\rangle_{\text{ideal}} = \cos(m\theta) - 1 \approx -(m^2\theta^2)/2 + \mathcal{O}(\theta^4)$; from Eq. (2.5), the twirled noise channel gives $\langle Z\rangle_{\text{noisy}} - \langle Z\rangle_{\text{ideal}} = [\cos(\theta)]^m - 1 \approx -(m\theta)/2 + \mathcal{O}(\theta^4)$. We see that coherent error builds up quadratically in the repetition parameter m, whereas stochastic Pauli error only builds up linearly in m.

We remark that single-qubit Clifford twirling is also commonly employed. Instead of sampling the twirling gates from the Pauli group, it samples from the 24-element single-qubit Clifford group. The resultant averaging effect creates further symmetry along the diagonal of the PTM[3].

This randomization technique can also be used to perform measurement readout mitigation. After applying a twirling gate right before the measurement, there is no physical twirling conjugation gate after the measurement. Instead, we virtually apply the conjugation gate by flipping the measurement result accordingly, shall the twirling gate inserted be X or Y (measured in the computational basis). The measurement twirling is part of the model-free twirled readout error extinction (TREX) technique [76].

Probabilistic Error Cancellation

Probabilistic error cancellation (PEC) [71, 72, 68] operates by intentionally applying a set of "inverse" errors to the quantum system, which can cancel out the natural errors occurring in a quantum computer on average. This process involves first characterizing the errors in the quantum system, and then, during computation, randomly applying these characterized error inverses according to a specific probability distribution. The result is a series of quantum operations that, on average, counteract the effect of the natural errors. It requires the noise to be stochastic Pauli error channel—for the convenience of characterization and also cancellation by insertion of Pauli gates—which requires noise tailoring techniques such as randomized compiling as described in Sec. 2.1.

The learning of the stochastic Pauli channel is usually done by techniques such as cycle benchmarking (CB) [77][4], which extracts Pauli fidelities before converting them to Pauli error probabilities through Walsh-Hadamard transformation [78, 79]. The learning requires

[3]The diagonals of the PTM are related to the Pauli fidelities λ_b defined in $\Lambda(\rho) = \sum_{b\in\mathbb{Z}_n^{2n}} \lambda_b \mathrm{Tr}(P_b\rho)P_b$. Single-qubit Clifford twirling forces all Pauli's with the same weight on the same qubits to have the same Pauli fidelities, e.g., $\lambda_{XIY} = \lambda_{ZIZ}$

[4]Randomized benchmarking (RB), on the other hand, employs up to two-qubit Clifford twirling for up to a set of two-qubit gates, which symmetrizes all Pauli fidelities, yielding a global average quantity of a fully-depolarizing rate (all Pauli fidelities the same, and thus all Pauli error probabilities are the same).

a substantial overhead, but may be reduced exponentially to a scaling of only linear in the number of qubits [78].

The idea of the noise inversion process in PEC (assuming the stochastic Pauli noise channel has been learned) can be illustrated in a simple single-qubit circuit with only X flip with a probability p. Suppose we randomly insert an X gate with a probability of q to counter the natural error, there are four scenarios: there is a probability of $(1-q)(1-p)$ that there was no error and we inserted no gate; there is a probability of $(1-q)p$ that there was an error but we inserted no gate; there is a probability of $q(1-p)$ that there was no error but we inserted a gate; and there is a probability of qp that there is an error canceled by the gate we inserted. Namely, there is a probability $(1-q)p+q(1-p)$ that we end up with an X error in the circuit. We wish this probability to vanish, which can be readily shown that it requires $q=-p/(1-2p)$. This can be a negative probability, or quasi-probability, since a noise channel is in general non-invertible (or to say that the inverted channel is non-physical). The sign of the quasi-probability can be taken care in the weighted average of the expectation values from the mitigation circuit, and we just need a relative magnitude of the probability to insert the noise cancellation gate. To this end, we re-normalize the probability of inserting a gate to be $|q|/(|1-q|+|q|)$. Let $\mathrm{Tr}[O\tilde{\mathcal{C}}(\rho)]$ represent the expectation value when we insert no gate and $\mathrm{Tr}[OX\tilde{\mathcal{C}}(\rho)X]$ represent the expectation value when we insert an X noise cancellation gate, the mitigated expectation value can be written as

$$\mathrm{Tr}[O\mathcal{C}(\rho)] = \frac{1}{|1-q|+|q|}\left\{\mathrm{sgn}(1-q)|1-q|\,\mathrm{Tr}[O\tilde{\mathcal{C}}(\rho)] + \mathrm{sgn}(q)|q|\,\mathrm{Tr}[OX\tilde{\mathcal{C}}(\rho)X]\right\}. \tag{2.6}$$

The above argument can be extended to having all Pauli errors. The PEC is done layer by layer, and thus it scales with the circuit depth.

Intuitively, the stochastic Pauli noise can be thought of as a random walk, and the noise cancellation by probabilistically inserting Pauli gates is another random walk process, both of which combine to yield an unbiased estimator of the expectation value at the expense of increased variance of the combined stochastic processes. Therefore, to control the variance of this unbiased estimator, a significant overhead is demanded, which is in fact exponential in the circuit size (circuit depth × width) as well as in the noise strength [68], making it practically challenging to implement.

Zero-noise Extrapolation

Zero-noise Extrapolation (ZNE) [71, 72] operates on the principle of deliberately introducing varying levels of noise into a quantum system and then performing the same quantum computation under these different noise conditions. By analyzing how the results of the computation change with different noise intensities, ZNE allows for the extrapolation of what the result would be in the absence of noise, hence "zero-noise". Since obtaining the actual extrapolating functional relationship under realistic noise is no easier than fully characterizing all the noise processes and/or performing the quantum computation, the true extrapolating

functional is unknown, and any extrapolating functional used in practical (usually polynomials) results in a biased estimator of the noise-free result.

There are various methods to introduce varying levels of noise into a quantum circuit. One is to fold the gates. For instance, noise level 3 means amplifying each gate's noise by a factor of 3, which can be achieved by performing $UU^\dagger U$ for each gate (usually only for each entangling gate). We can also do a fractional noise level, meaning a fraction of the gates have noise their amplified (or equivalently gate folded). This is called digital ZNE. Another approach amplifies the gate noise by stretching the pulse (extending its duration) of the gate by the same factor. This is called analog ZNE or pulse-stretching ZNE. The last one is to convert the gate noise into stochastic Pauli noise by randomized compiling and then increasing the probability of the Pauli noise by some factor through manually inserting Pauli gates—a process similar to PEC but instead of inserting to cancel, here we insert to amplify. This is called probabilistic error amplification (PEA) [70].

Virtual Distillation

Many quantum algorithms target the preparation of a pure ideal state $\rho = |\psi\rangle\langle\psi|$. Many common noise channels are stochastic, which will turn our ideal pure state ρ into some noisy mixed state. Virtual distillation (VD) [80, 81] aims to extract the eigenvector corresponding to the largest eigenvalues of ρ, since the closest pure state to a ρ in the trace distance is simply the dominant eigenvector $|\phi\rangle\langle\phi|$.

This can be elegantly achieved by raising the density matrix to a sufficiently large power so that the dominant eigenvector distinguishes itself sufficiently. Namely, we would like to achieve this mapping

$$\rho \mapsto \frac{\rho^m}{\mathrm{Tr}\left(\rho^m\right)}. \tag{2.7}$$

It has been shown elegantly by e.g., tensor diagrams, that measuring a ρ to a power m is equivalent to measuring a local observable of m copies in parallel acted by a cyclic permutation operator [80]

$$\mathrm{Tr}(O\rho^m) = \mathrm{Tr}(S_m O_k \rho^{\otimes m}), \tag{2.8}$$

where O_k is the observable of interest on the k-th copy of the noisy ρ, and S_m is the cyclic permutation operator performing $S_m |\psi_1\rangle |\psi_2\rangle \dots |\psi_m\rangle = |\psi_2\rangle |\psi_3\rangle \dots |\psi_1\rangle$.

Numerical and analytic studies have found that the error mitigation from VD can be of multiple orders of magnitude for large systems, even using as little as $m = 2$ copies of the state [80, 81]. However, the implementation of the cyclic permutation operator is hindered by NISQ device connectivity, since it can require long-range entangling gates.

2.2 Machine Learning for Quantum Error Mitigation

Emerging at the crossroads of quantum mechanics and statistical learning, machine learning for quantum error mitigation (ML-QEM) presents a promising avenue where statistical mod-

els are trained to derive mitigated expectation values from noisy counterparts executed on quantum computers. Could such ML-QEM methods offer valuable improvement in accuracy or runtime efficiency in practice?

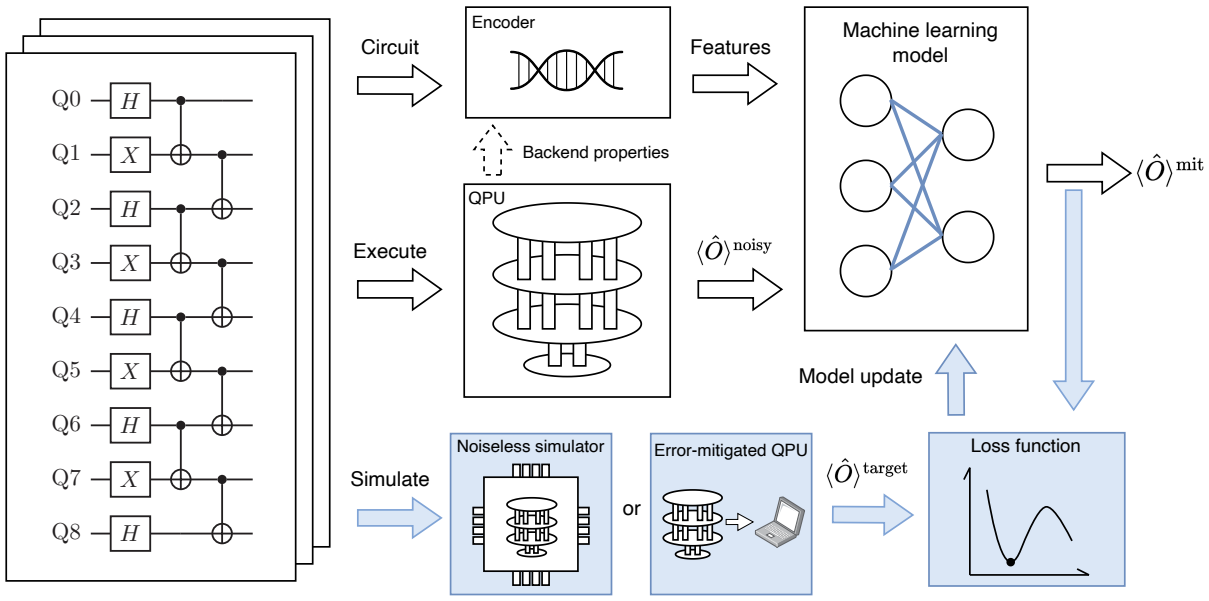


Figure 2.1: Machine-learning quantum error mitigation (ML-QEM): execution and training for tractable and intractable circuits. A quantum circuit (left) is passed to an encoder (top) that creates a feature set for the ML model (right) based on the circuit and the quantum processor unit (QPU) targeted for execution. The model and features are readily replaceable. The executed noisy expectation values $\langle\hat{O}\rangle^{\text{noisy}}$ (middle) serve as the input to the model whose aim is to predict their noise-free value $\langle\hat{O}\rangle^{\text{mit}}$. To achieve this, the model is trained beforehand (bottom, blue highlighted path) against target values $\langle\hat{O}\rangle^{\text{target}}$ of example circuits. These are obtained either using noiseless simulations in the case of small-scale, tractable circuits or using the noisy QPU in conjunction with a conventional error mitigation strategy in the case of large-scale, intractable circuits. The training minimizes the loss function, typically the mean square error. The trained model operates without the need for additional mitigation circuits, thus reducing runtime overheads.

In principle, a successful ML-QEM strategy would learn the effect of noise in training, thus obviating the need for additional mitigation circuits during the execution of an algorithm. Compared to conventional QEM, the algorithmic runtime would then see a potential reduction in overhead. First explorations of ML-QEM ideas have shown signs of promise, even for complex noise profiles [82, 83, 84, 85, 86, 87], but it remains unclear if ML-QEM can perform in practice in quantum computations on hardware or at scale. For instance, it is unclear whether a given ML-QEM method can be used across different device noise profiles, diverse circuit classes, and large quantum circuit volumes beyond the limits of classical

simulation. To date, there has not been a systematic study comparing different traditional methods and statistical models for QEM on equal footing under practical scenarios across a variety of relevant quantum computational tasks.

In this study, we present a general framework to perform ML-QEM for higher runtime efficiency compared to other mitigation methods. Our study encompasses a broad spectrum of simple to complex machine learning models, including the previously proposed linear regression and multi-layer perceptron model. We further propose two new models, random forests and graph neural networks. We find that random forests seem to consistently perform the best. We evaluate the performance of all four models in diverse realistic scenarios. We consider a range of circuit classes (random circuits and Trotterized Ising dynamics) and increasingly complex noise models in simulations (including incoherent and coherent gate errors, readout errors, and qubit errors). Additionally, we explore the advantages of ML-QEM methods over traditional approaches in common use cases, such as generalization to unseen Pauli observables, and enhancement of variational quantum-classical tasks. Our analysis reveals that ML-QEM methods, particularly random forest, exhibit competitive performance compared to a state-of-the-art method—digital zero-noise extrapolation (ZNE)—while requiring lower overhead by a factor of at least 2 in runtime. Finally, with experiments on IBM quantum computers for quantum circuits with up to 100 qubits and a two-qubit gate depth of 40, we propose a path toward scalable mitigation by mimicking traditional mitigation methods with superior runtime efficiency, which also serves as a further example of using classical machine learning on quantum data [88, 89, 1].

2.3 Simulations and Hardware Experiments

The ML-QEM workflow (see Fig. 2.1) operates on a given class of quantum circuits for which we train an ML model to predict near noise-free expectation values based on noisy expectation values obtained from a quantum processing unit (QPU). This is required, since, in general, the output of the quantum circuit is considered intractable and cannot be learned in isolation by the ML model. Details of the training set, encoded circuit and QPU features, and ML models, can be found in the Methods section. A key feature of the ML-QEM model is that at runtime, the model produces mitigated expectation values from the noisy ones without the need for additional mitigation circuits, thus dramatically reducing overheads.

As reported in the following, we find at par or even improved accuracy results at significantly reduced runtime overheads using machine learning approaches in reference to the chosen reference digital gate-folding ZNE approach[5][6]. We first report on performance in small-scale circuits trained and tested in numerical simulations under realistic noise models. In turn, we validate conclusions on real noisy hardware. We then introduce a scalable

[5]We use zero-noise extrapolation with digital gate folding on 2-qubit gates, noise factors of $\{1, 3\}$, and linear extrapolation implemented via Ref. [90].

[6]All non-ideal expectation values in simulations and experiments presented in this study are obtained from the measurement statistics from 10,000 shots.

ML-QEM methodology for large-scale circuits. We demonstrate this idea in an experiment using 100-qubit circuits with up to 1,980 CNOT gates. This is accomplished by training the ML models to mimic the results of conventional error mitigation methods, inheriting their accuracy but with significantly reduced overhead. In this regime, the calculations are beyond simple brute-force numerical techniques, and serve as a test-bed for intractable circuits.

Mitigating Depolarizing Noise

Before presenting our simulations and experiments, we motivate the use of noisy expectation values in the input features to the ML-QEM, and show here that the ideal expectation values of an observable $\hat{O}$ linearly depend on its noisy expectation values when the noisy channel of the circuit consists of successive layers of depolarizing channels. This is more general than the result shown in [83].

Consider l successive layers of unitaries each associated with a depolarizing channel with some rate p_l, the noisy circuit acting on the input ρ, $\tilde{\mathcal{C}}(\rho)$, is written as $\tilde{\mathcal{C}}(\rho) = \mathcal{E}_l(U_l\mathcal{E}_{l-1}(U_{l-1}\ldots\mathcal{E}_1(U_1\rho U_1^\dagger)\ldots U_{l-1}^\dagger)U_l^\dagger)$, where $\mathcal{E}_l(\rho) = (p_l/D)I + (1-p_l)\rho$.

It can be shown by induction that

$$\tilde{\mathcal{C}}(\rho) = \frac{p(l)}{D}I + (1-p(l))U_l\ldots U_1\rho U_1^\dagger\ldots U_l^\dagger, \tag{2.9}$$

where $p(l) = 1 - \Pi_{i=1}^{l}(1-p_i)$ as follows. Assuming for $l = k$,

$$\tilde{\mathcal{C}}(\rho) = \frac{p(k)}{D}I + (1-p(k))U_k\ldots U_1\rho U_1^\dagger\ldots U_k^\dagger, \tag{2.10}$$

then for $l = k+1$, we have

$$\begin{aligned}\tilde{\mathcal{C}}(\rho) &= \frac{p(k)}{D}I + (1-p(k))\left[\frac{p_{k+1}}{D}I + (1-p_{k+1})U_k\ldots U_1\rho U_1^\dagger\ldots U_k^\dagger\right] \\ &= \frac{p(k+1)}{D}I + (1-p(k+1))U_k\ldots U_1\rho U_1^\dagger\ldots U_k^\dagger.\end{aligned} \tag{2.11}$$

The induction completes with a trivial base case.

Therefore, the noisy expectation value of $\hat{O}$ becomes

$$\begin{aligned}\mathrm{Tr}(\tilde{\mathcal{C}}(\rho)\hat{O}) &= \frac{p(l)}{D}\mathrm{Tr}(\hat{O}) + (1-p(l))\mathrm{Tr}(U_l\ldots U_1\rho U_1^\dagger\ldots U_l^\dagger\hat{O}) \\ &= \frac{p(l)}{D}\mathrm{Tr}(\hat{O}) + (1-p(l))\mathrm{Tr}(\mathcal{C}(\rho)\hat{O})\ ,\end{aligned} \tag{2.12}$$

where $\mathrm{Tr}(\mathcal{C}(\rho)\hat{O})$ is the ideal expectation value of $\hat{O}$.

For Trotterized circuits with a fixed Trotter step and a fixed brickwork structure, the number of layers l of unitaries in the circuit is also fixed. Assuming some fixed-rate depolarizing channels associated with the l layers of unitaries, the noisy and ideal expectation values

of some $\hat{O}$ on these Trotterized circuits with different parameters then lie on a line. Therefore, the ML-QEM method can mitigate the expectation values by linear regression from the noisy expectation values to the ideal ones, and the linear regression parameters can be learned to vary according to the number of layers l. The ML-QEM is thus *unbiased* in this case. We note that ZNE with linear extrapolation is still *biased* in this case, since the noise amplification effectively results in a different combined depolarizing rate $p'(l) = 1 - \Pi_{i=1}^{l}(1-p'_i)$, which leads to expectation values with differently amplified noises each lying on a different line towards the ideal expectation value, and thus the linear extrapolation cannot yield unbiased estimates.

Performance Comparison at Tractable Scale

First, we present a comparative analysis of several representative ML-QEM methods. As portrayed in Fig. 2.9 in the Methods section in Sec. 2.4, we explore several statistical models in our study with varying complexity and methods of encoding data, namely linear regression with ordinary least squares (OLS), random forests (RF), multi-layer perceptrons (MLP), and graph neural networks (GNN). Since the relationship between the noisy expectation values and the ideal ones is non-linear in general (see Sec. 2.3 for more details), we emphasize the role of non-linear machine learning models, and study three non-linear models, i.e., RF, MLP, and GNN, in addition to the linear model OLS. Each of these models is described in further detail in the Methods. We compare these models against each other and digital gate-folding ZNE. Future studies comparing ML-QEM against methods with more rigorous theoretical guarantees and successful experimental demonstrations, such as probabilistic error cancellation [68], probabilistic error amplification [70], and analog zero-noise extrapolation *via* pulse stretching [69] are warranted, as digital gate-folding ZNE is known to be accurate only under depolarizing noise models [91].

We evaluate the performance of these methods for two classes of circuits: random circuits and Trotterized dynamics of the 1D Ising spin chain on small-scale simulations. These two classes of circuits bear distinct two-qubit gate arrangements, allowing us to gain knowledge about the performance of the ML-QEM on the two extremes of the spectrum in terms of circuit structures. This evaluation is done by simulations on small-scaled circuits, conveniently allowing us to vary the type of noises affecting the circuits and to identify situations under which the ML-QEM outperforms digital ZNE in terms of mitigation accuracy. To that end, in the study of Trotterized circuits, we also study the performance of the methods in the absence and presence of readout error or coherent noise, in addition to incoherent noise.

Random Circuits

In the first experiment, we benchmark the performance of the protocol on small-scale unstructured circuits. To ensure that the circuits encompass a broad spectrum of complexities, we generate a diverse set of four-qubit random circuits with varying two-qubit gate depths, up to a maximum of 18, as shown in the inset of Fig. 2.2. Per two-qubit depth, there are

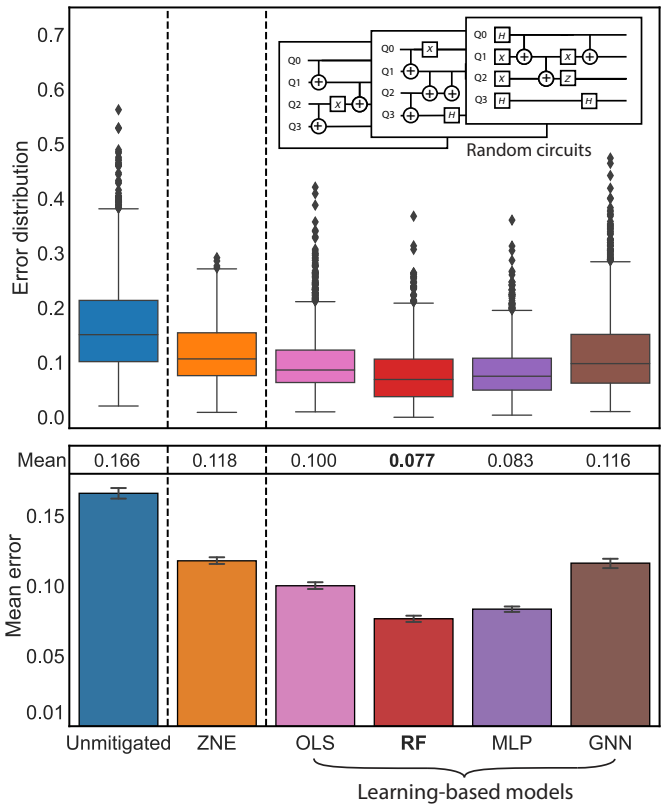


Figure 2.2: Quantum error mitigation (QEM) and ML-QEM accuracy on random circuits. Top: Error distribution for unmitigated and mitigated Pauli-Z expectation values. Mitigation is performed using either a reference QEM method, digital zero-noise extrapolation (ZNE), or one of four ML-QEM models (explained in text). Inset: Example random circuits. Noisy execution is numerically simulated using a noise model derived from IBM QPU `Lima`. The error is defined as the L_2 distance between the vector of all ideal and noisy single-qubit expectations $\langle \hat{Z}_i \rangle$; i.e., $\| \langle \hat{Z} \rangle - \langle \hat{Z} \rangle_{\text{ideal}} \|_2$. Black dots are outliers. Average is over 2,000 four-qubit random circuits, with two-qubit-gate depths sampled up to 18. Bottom: Average error for each method (using data from the top) is presented with 95% confidence intervals, derived from bootstrap re-sampling. The mean L_2 error is provided above each column.

500 random training circuits and 200 random test circuits that are generated by the same sampling procedure. For each circuit, we carry out simulations on IBM's `FakeLima` backend, which emulates the incoherent noise present in the real quantum computer, the `ibmq_lima` device. While these quantum devices generally have coherent errors as well, they can be suppressed through a combination of e.g., dynamical decoupling [92, 93] and randomized compiling [68, 75]. Specific types of noise include incoherent gate errors, qubit decoherence, and readout errors. We train the ML-QEM models to mitigate the noisy expectation values of the four single-qubit $\hat{Z}_i$ observables. As a benchmark, we also compare mitigated

expectation values from digital ZNE. In Fig. 2.2, we show the error (between the mitigated expectation values and the ideal ones) distribution of digital ZNE and ML-QEM with each of the four machine learning models on the top plot and the bootstrap mean errors in the bottom plot. We observe that the random forest consistently outperforms the other ML-QEM models, with the MLP model closely following. Notably, all ML-QEM models, including OLS and GNN, exhibit competitive performance in comparison to the ZNE method, despite that the runtime overhead for ZNE is twice as much. Finally, we emphasize that rigor hyperparameter optimization may impact the relative performance of these methods, and we leave this analysis to future work.

We remark that in the study of 4-qubit random circuits presented in this section, we use the Qiskit function `qiskit.circuit.random.random_circuit()` to generate the random circuits, which implements random sampling and placement of 1-qubit and 2-qubit gates, with randomly sampled parameters for any selected parametrized gates. The 2-qubit gate depth is measured after transpilation. We remark that the random circuits sampled at large depths may approximate the Haar distribution and have expectation values concentrated around 0 to some extent [94, 95].

Trotterized 1D Transverse-field Ising Model

To benchmark the performance of the protocol on structured circuits, we consider Trotterized brickwork circuits. Here, we consider first-order Trotterized dynamics of the 1D transverse-field Ising model (TFIM) subject to different noise models based on the incoherent noise on the `FakeLima` simulator in Fig. 2.3, before moving to experiments on IBM hardware with actual device noise in Fig. 2.4. We observe that these circuits are not only broadly representative but also bear similarities to those used for Floquet dynamics [96]. The dynamics of the spin chain is described by the Hamiltonian

$$\hat{H} = -J \sum_j \hat{Z}_j \hat{Z}_{j+1} + h \sum_j \hat{X}_j = -J\hat{H}_{ZZ} + h\hat{H}_X \ , \tag{2.13}$$

where J denotes the exchange coupling between neighboring spins and h represents the transverse magnetic field, whose first-order Trotterized circuit is shown in the inset of Fig. 2.3. We generate multiple instances of the problem with varying numbers of Trotter steps and coupling strengths, such that the coupling strengths of each circuit are uniformly sampled from the paramagnetic phase ($J < h$) by choice. There are 300 training circuits and 300 testing circuits per Trotter step, and the training circuits cover Trotter steps up to 14. Each circuit is measured in a randomly chosen Pauli basis for all the weight-one observables. We then train the ML-QEM models on the ideal and noisy expectation values obtained from these circuits and compare their performance with digital ZNE. During the testing phase, we consider both interpolation and extrapolation. In interpolation, we test on circuits with sampled coupling strength J not included in training but with Trotter steps included in the training. In extrapolation, we test on circuits with sampled coupling strength J not

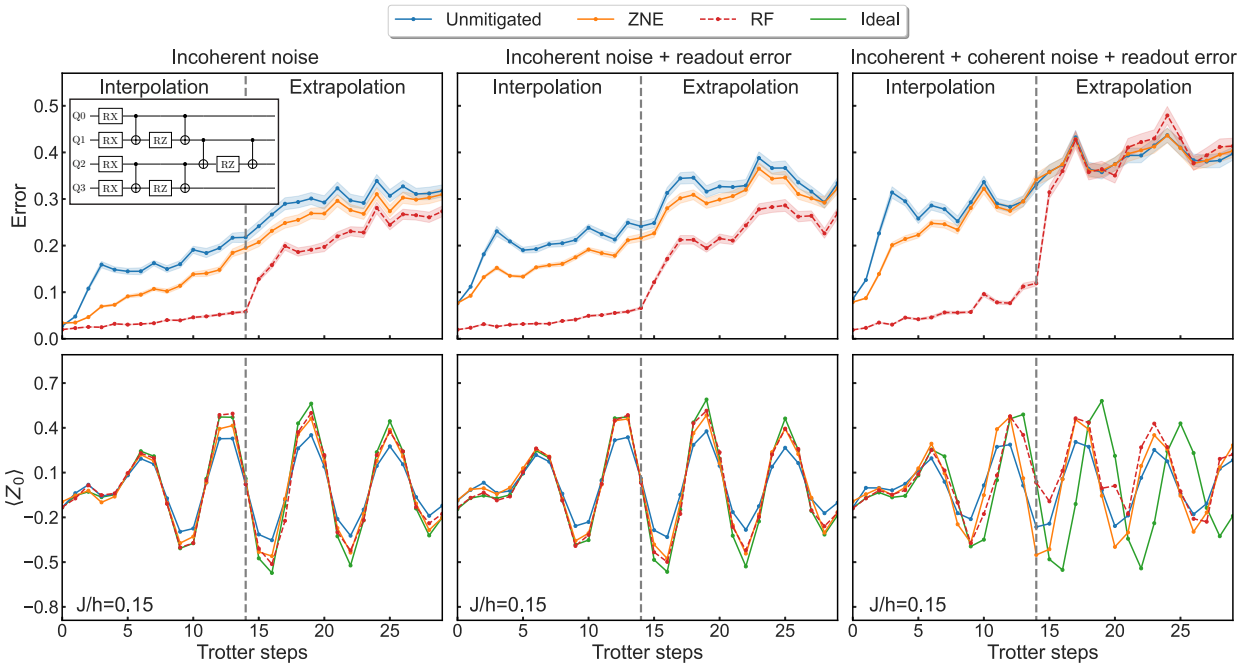


Figure 2.3: Mitigation accuracy under i) complexity of quantum noise and ii) ML-QEM interpolation and extrapolation for Trotter circuits. Top row: Average error performance on Trotter circuits (top-left inset) representing the quantum time dynamics of a four-site, 1D, transverse-field Ising model in numerical simulations. A Trotter step comprises four layers of CNOT gates (inset). Vertical dashed line separates experiments in the ML-QEM interpolation regime (left) from the extrapolation regime (right). The 3 curves represent the performance of the highest-performing ML-QEM method, the QEM ZNE method, and the unmitigated simulations. They are averaged over 300 circuits, each with a randomly chosen Pauli measurement bases. The data is for all four weight-one expectations $\langle \hat{P}_i \rangle$. The error is defined as L_2 distance from the ideal expectations, $\|\langle \hat{P} \rangle - \langle \hat{P} \rangle_{\text{ideal}}\|_2$, as also defined for the remainder of figures. From the left to right, the complexity of the device noise model increases to include additional realistic noise types. Coherent errors are introduced on CNOT gates. Bottom row: Corresponding typical data of the error-mitigated expectation values of the $\langle Z_0 \rangle$ Trotter evolution; here, for Ising parameter ratio $J/h = 0.15$.

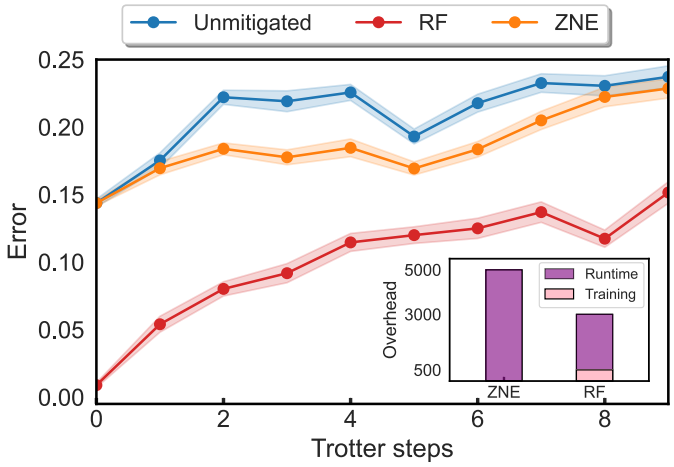


Figure 2.4: On QPU hardware: accuracy and overhead for ML-QEM and QEM. Average execution error of Trotter circuits for experiments on QPU device `ibm_algiers` without mitigation and with ZNE or ML-QEM RF mitigation. Error performance is averaged over 250 Ising circuits per Trotter step, each with sampled Ising parameters $J < h$ and each measured for all weight-one observables in a randomly chosen Pauli basis. Training is performed over 50 circuits per Trotter step, which results in both a 40% lower *overall* and 50% lower *runtime* quantum resource overhead of RF compared to the overhead of the digital ZNE implementation (see inset).

included in the training as well as with Trotter steps exceeding the maximal steps present in the training circuits.

On the noisy simulator in Fig. 2.3, for this problem with incoherent gate noise in the absence (left) or presence (right) of readout error, the ML-QEM model (using the random forest) performs better than the ZNE method. We envision that ML-QEM can be used to improve the accuracy of noisy quantum computations for circuits with gate depths exceeding those included in the training set.

On the right of Fig. 2.3, we consider the same problem in the second study but simulate the sampled circuits on `FakeLima` backend with additional coherent errors. The added coherent errors are CNOT gate over-rotations with an average over-rotational angle of 0.02π. We again generate multiple instances of the problem with varying numbers of Trotter steps and coupling strengths uniformly sampled from the paramagnetic phase. We then train the ML-QEM model (using the random forest) on the ideal and noisy expectation values obtained from these circuits executed on the modified `FakeLima` backend and compare their performance with ZNE.

During the testing phase, we also perform extrapolation where some testing circuits have Trotter steps exceeding the maximal steps present in the training circuits. Specifically, the testing circuits cover 14 more steps up to Trotter step 29. Under the influence of added

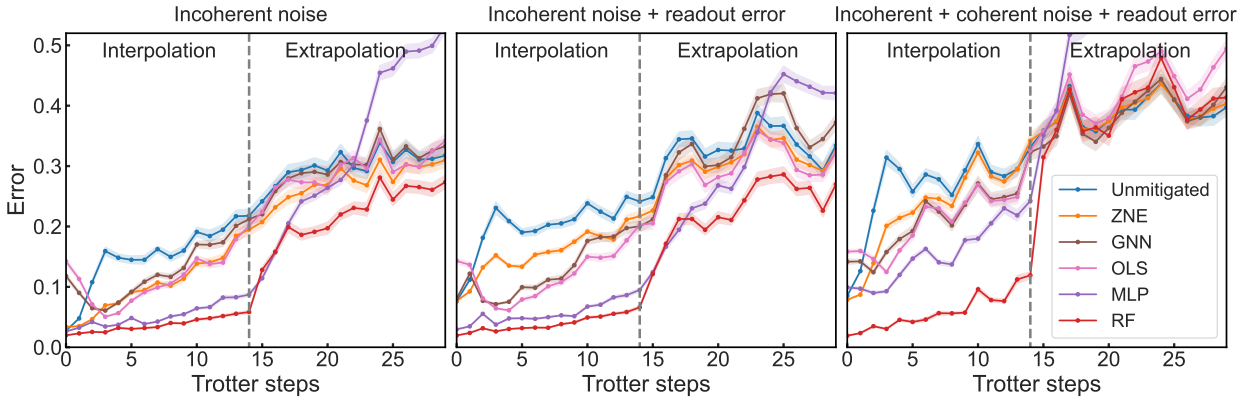


Figure 2.5: ML-QEM and QEM performance for Trotter circuits. Expanded data corresponding to Fig. 2.3 of the main text that includes the three ML-QEM methods not shown earlier: GNN, OLS, MLP. We study three noise models: Left: incoherent noise resembling `ibmq_lima` without readout error, Middle: with the additional readout error, and Right: with the addition of coherent errors on the two-qubit CNOT gates. We show the depth-dependent performance of error mitigation averaged over 9,000 Ising circuits, each with different coupling strengths J. For the incoherent noise model, all ML-QEM methods demonstrate improved performance even when mitigating circuits with depths larger than those included in the training set. However, all perform as poorly as the unmitigated case in extrapolation with additional coherent noise.

coherent noise, the performance of the ML-QEM model and digital ZNE deteriorated compared to the previous study. However, in the extrapolation scenario, none of the models demonstrated effective mitigation of the noisy expectation values. In practical applications, a combination of, e.g., dynamical decoupling [92] and randomized compiling [68, 75], which can suppress all coherent errors, could be applied to the test circuits prior to utilizing ML-QEM models. This approach effectively converts the noise into incoherent noise, enabling the ML-QEM methods to perform optimally in extrapolation. We remark that coherent gate errors induce quadratic changes in the expectation values, which are stronger than incoherent errors inducing only linear changes—it is plausible that the machine learning approach performs better with weak noises.

We present a comparison across all ML-QEM models in this study of mitigating expectation values of Trotterized 1D TFIM in Fig. 2.5. With incoherent noise only, the random forest model demonstrates the best performance among the ML-QEM models both in interpolation and extrapolation, closely followed by the MLP, OLS, and GNN. With additional coherent noise, in the interpolation scenario, the performance ranking of the other models remained largely consistent with that observed in the previous study. Notably, the random forest model exhibited the best performance among the ML-QEM models, closely followed by the MLP model.

We benchmark the performance of the ML-QEM model against digital ZNE on real quantum hardware, IBM's `ibm_algiers`. In this experiment, we do not apply any additional error suppression or error mitigation such as dynamical decoupling, randomized compiling, or readout error mitigation; thus, the experiment involves incoherence noise, coherent noise, and readout error, with the results shown in Fig. 2.4. We train the ML-QEM with random forest on 50 circuits and test it on 250 circuits at each Trotter step. We observe that 50 training circuits per step, totaling 500 training circuits, suffices to have the model trained well. With this low train-test split ratio[7], the ML-QEM requires $500 + 2{,}500 = 3{,}000$ total circuits, while running ZNE with 2 noise factors on the testing circuits requires $2 \times 2{,}500 = 5{,}000$ total circuits. The ML-QEM claims a reduction of quantum resource overhead compared to ZNE *both overall and at runtime*—the reduction is as large as 30% overall and 50% at runtime. Additionally, we observe that the ML-QEM method RF significantly outperforms ZNE for all Trotter steps, demonstrating the efficacy of this approach under a realistic scenario. We report approximately 0.7 QPU hours (based on a conservative sampling rate of 2 kHz [70]) to generate all the training data and seconds to train the model with a single-core laptop for this experiment.

We remark that in this study of the Trotterized 1D TFIM, we initialize the state devoid of spatial symmetries. This is done to intentionally introduce asymmetry in the single-qubit $\hat{Z}_i$ expectation value trajectories across Trotter steps, thereby increasing the difficulty of the regression task. Conversely, when the initial state possesses a certain degree of symmetry, the regression analysis, which incorporates noisy expectation values as features, becomes highly linear, resulting in a strong performance by the OLS method.

We observe that both in the simulation and in the experiment of the small-scale Trotterized 1D TFIM, there are significant correlations between the noisy expectation values and the ideal ones. There are also significant correlations but to a lesser degree between the gate counts and the ideal expectation values, suggesting the models are using certain depth information deduced from the gate counts to correct the noisy expectation values towards the ideal ones.

Mitigating Unseen Pauli Observables

There are algorithms in which we care about the expectation values of multiple non-commuting Pauli observables on the same circuit, effectively creating multiple target circuits with the same gate sequences but with different measuring basis, such as in quantum state tomography and in variational quantum eigensolver. Additional error mitigation methods incur a large overhead on top of these target circuits by requiring additional mitigation circuits for

[7] Assuming the mimicked QEM requires m total executions of either the mitigation circuits or the circuit of interest (e.g., digital/analog ZNE usually has $m = 2$ or 3 noise factors), the total cost of the mimicked QEM, namely its runtime cost, is mn_{test}. The total cost, including training, for the RF is $mn_{\text{train}} + n_{\text{test}}$. Equating these two yields the break-even train-test split ratio in the total cost of our mimicry compared to the traditional QEM: $n_{\text{train}}/n_{\text{test}} = (m-1)/m$. Our mimicry shows a higher overall efficiency when the train-test split ratio is smaller than $(m-1)/m$.

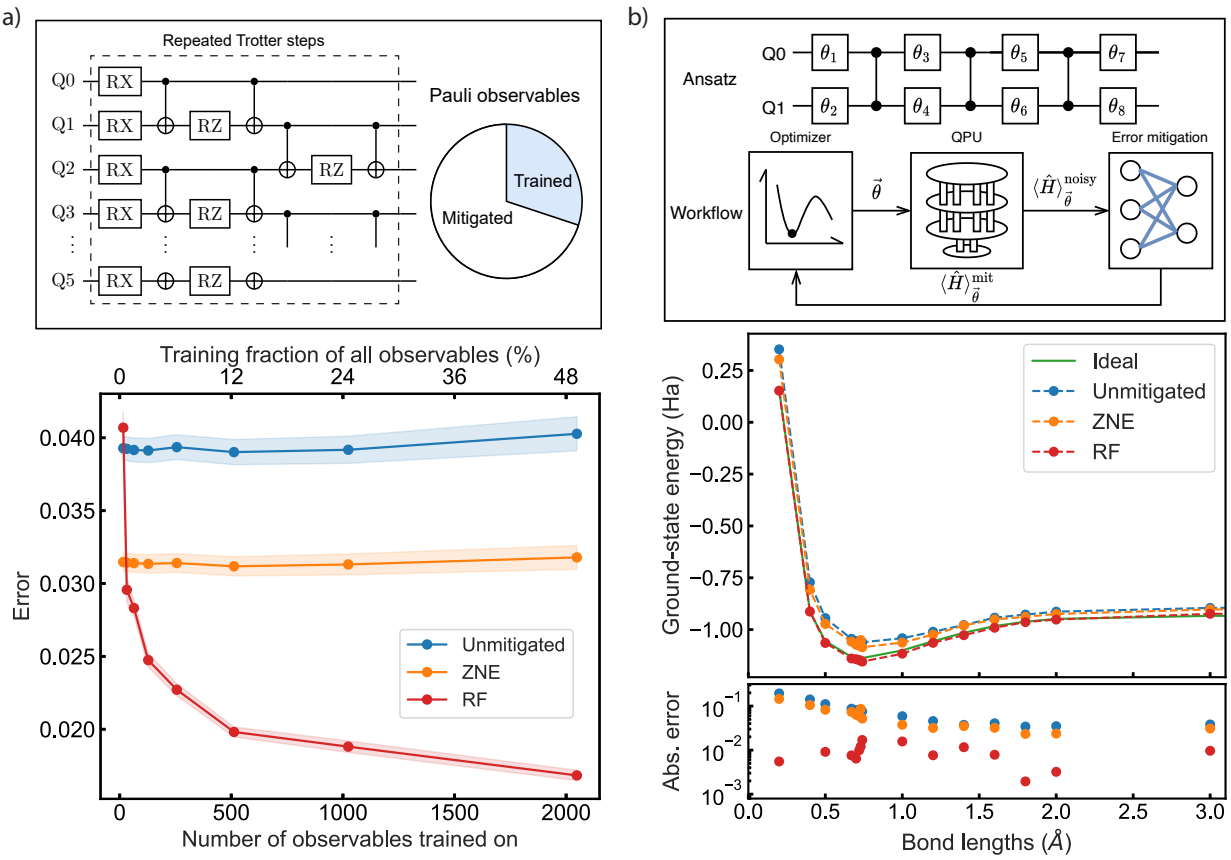


Figure 2.6: Application of ML-QEM to a) unseen expectation values and b) the variational quantum eigensolver (VQE). a) Top: Schematic of a Trotter circuit, which prepares a many-body quantum state on $n = 6$ qubits (in 5 Trotter steps). Top right: Circle depicts the pool of all possible 4^n Pauli observables. Shaded region depicts the fraction of observables used in training the ML model; the remaining observables are unseen prior to deployment in mitigation. Bottom: Average error of mitigated unseen Pauli observables versus the total number of distinct observables seen in training. b) Top: Schematic of the VQE ansatz circuit for 2 qubits parametrized by 8 angles $\vec{\theta}$. Below, a depiction of the VQE optimization workflow optimizing the set of angles $\vec{\theta}$ on a simulated QPU, yielding the noisy chemical energy $\langle \hat{H} \rangle_{\vec{\theta}}^{\text{noisy}}$, which is first mitigated by the ML-QEM or QEM before being used in the optimizer as $\langle \hat{H} \rangle_{\vec{\theta}}^{\text{mit}}$. Compared to the ZNE method, the ML-QEM with RF method obviates the need for additional mitigation circuits at every optimization iteration at runtime.

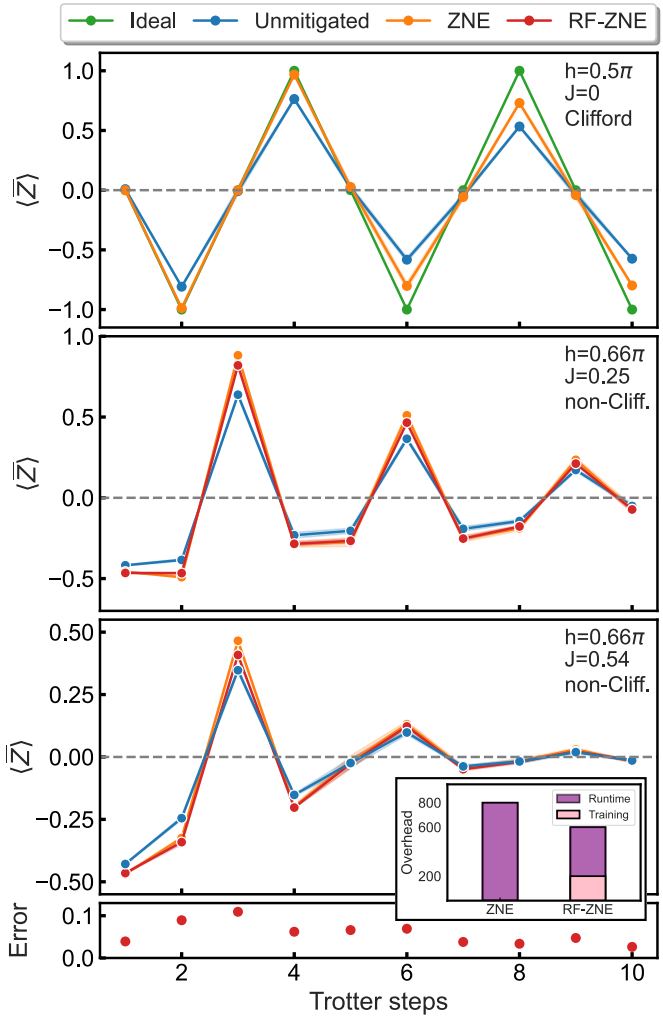


Figure 2.7: ML-QEM mimicking QEM on large, 100-qubit circuits with lower overheads, in hardware. Top three panels: Average expectation values from 100-qubit Trotterized 1D TFIM circuits executed in hardware on QPU `ibm_brisbane`. Each panel corresponds to a different Ising parameter set (top right corners). Top panel corresponds to a Clifford circuit, whose ideal, noise-free expectation values are shown as the green dots. The RF-mimicking-ZNE (RF-ZNE) curve corresponds to training the RF model against ZNE-mitigated data on the hardware rather than in numerical simulators, for which these large non-Clifford circuits are more difficult. Bottom panel: The error, measured again in the L_2 norm, between the ZNE-mitigated expectation values and the RF-mimicking-ZNE (RF-ZNE) mitigated expectation values over non-Clifford testing circuits with randomly sampled coupling strengths $J < h$ averaged over 40 testing circuits per Trotter step and the observables. The training is over 10 circuits per Trotter step, which results in a 25% lower *overall* and 50% lower *runtime* quantum resource overhead compared to the ZNE, as shown in the inset.

each target circuit. Here, we show that it is possible to achieve better mitigation performance with lower overhead using an ML-QEM method.

In particular, we evaluate the performance of the ML-QEM to mitigate unseen Pauli observables on a state $|\psi\rangle$ produced by the Trotterized Ising circuit depicted on the top of Fig. 2.6(a), which contains 6 qubits and 5 Trotter steps. We train the random forest model on increasing fractions of the $4^6 - 1 = 4{,}095$ Pauli observables of a Trotterized Ising circuit with $J/h = 0.15$, and then we apply the model to mitigate noisy expectation values sampled from the *rest* of all possible Pauli observables. The results of this study are plotted at the bottom of Fig. 2.6(a). We observe that training the random forest on just a small fraction ($\lesssim 2\%$) of the Pauli observables results in mitigated expectation values with errors lower than when using ZNE. The ML-QEM method additionally has lower runtime overhead—there are no mitigation circuits with amplified noise required at runtime, and the number of circuits needed to be executed at runtime for the ML-QEM is at least a factor of 2 fewer than that for digital ZNE.

Enhancing Variational Algorithms

In the conventional formulation of the variational quantum eigensolver (VQE) algorithm, the goal is to estimate the ground-state energy by measuring the energy $\langle \hat{H} \rangle_{\vec{\theta}}$ of the state prepared by a circuit ansatz $\hat{U}(\vec{\theta})$ with a fixed structure and parameters $\vec{\theta}$. Then, a classical optimizer is used to propose a new $\vec{\theta}$, and this procedure is executed repeatedly until $\langle \hat{H} \rangle_{\vec{\theta}}$ converges to its minimum. When executing this algorithm on a noisy quantum computer, error mitigation can be used to improve the noisy energy $\langle \hat{H} \rangle_{\vec{\theta}}^{\text{noisy}}$ to the mitigated energy $\langle \hat{H} \rangle_{\vec{\theta}}^{\text{mit}}$ and better estimate the ground-state energy. This workflow is shown at the top of Fig. 2.6(b). Error-mitigated VQE with traditional methods can be costly, however, as additional mitigation circuits must be executed during each iteration. We use ML-QEM error mitigation instead, where a model is trained beforehand to mitigate the ground-state energy of an ansatz $\hat{U}(\vec{\theta})$ so that at each iteration, no additional mitigation circuits need to be executed. This approach for faster error mitigation at runtime may be especially appropriate for this algorithm, as VQE can require many iterations and long execution times during which quantum hardware can drift. A trained model could also then be used for error-mitigated VQE for different Hamiltonians.

To demonstrate this concept, we train the ML-QEM model with RF on 2,000 circuits with each parameter randomly sampled from $[-5, 5]$, and compute the dissociation curve of the H_2 molecule on the bottom of Fig. 2.6(b). The ML-QEM random forest model is trained on a two-local variational ansatz (depicted on the top of Fig. 2.6(b)) across many randomly sampled $\{\vec{\theta}\}$. This method results in energies that are close to chemical accuracy. Notably, the absolute errors are an order of magnitude smaller than those of the ZNE-mitigated energies.

Additionally, the training overhead of the ML-QEM model in VQE with different Hamiltonians can be significantly reduced by generalizing to mitigating unseen Pauli observables

in Sec. 2.3. By decomposing $\hat{H} = \sum_i c_i \hat{P}_i$ into Pauli terms, the ML-QEM only needs to train on the sampled ansatz $\hat{U}(\vec{\theta})$ with a *subset* of the Pauli observables in $\hat{H}$. This is demonstrated in our experiment shown in Fig. 2.6(b) where the random forest is trained on sampled ansatz $\hat{U}(\theta)$ measured in $\hat{X}_1\hat{X}_2$ and $\hat{Z}_1\hat{Z}_2$ (1,000 for each observable), while the Hamiltonian of the H_2 molecule at each bond length consists of $\hat{X}_1\hat{X}_2$, $\hat{Z}_1\hat{Z}_2$, $\hat{I}_1\hat{Z}_2$ and $\hat{Z}_1\hat{I}_2$ Pauli observables [97].

Scalability through Mimicry

For large-scale circuits whose ideal expectation values of certain observables are inefficient or impossible to obtain by classical simulations, we cannot train the model to mitigate expectation values towards the ideal ones. Rather, we could train the model to mitigate expectation values towards values mitigated by *other* scalable QEM methods, enabling scalability of ML-QEM through mimicry. Mimicry can be concretely visualized using the workflow for ML-QEM depicted in Fig. 2.1 with an *error-mitigated QPU* selected instead of a *noiseless simulator*, as we show in the inset of Fig. 2.7. Performing mimicry does not allow the ML-QEM model to outperform the mimicked QEM method by its nature, but allows the ML-QEM model to reduce the overhead compared to the traditional ML-QEM.

We demonstrate this capability by training an ML-QEM model to mimic digital ZNE in a 100-qubit Trotterized 1D TFIM experiment on `ibm_brisbane`. In particular, we use ZNE to mitigate five single-qubit $\hat{Z}_i$ observables on five qubits on the Ising chain with varying numbers of Trotter steps and J/h values. Each Trotter step contains 4 layers of parallel CNOT gates, and the circuits at Trotter step 10 have 1,500 CNOT gates in total. As shown in the top of Fig. 2.7, we first confirm that the ZNE-mitigated expectation values are more accurate than the unmitigated ones by benchmarking ZNE on a 100-qubit Trotterized Ising circuit with $h = 0.5\pi$ and $J = 0$ such that it is Clifford and classically simulable. We then train a random forest model to mitigate noisy expectation values the same way that ZNE does. In this experiment, we apply Pauli twirling to all the circuits, each with 5 twirls, before applying either extrapolation in digital ZNE or the ML-QEM to mitigate the expectation values.

We then find that the ML-QEM models are able to accurately mimic the traditional method's mitigated expectation values. The average distance from the unmitigated result (after twirling average) for the mitigated expectation values produced by ZNE and the random forest model mimicking ZNE are very close for all Trotter steps, as shown for specific J and h corresponding to non-Clifford circuits in the second and third panel of Fig. 2.7. In the fourth and bottom panel showing the residuals between the ZNE-mitigated and RF-mimicking-ZNE-mitigated values averaged over the training set comprising non-Clifford circuits, we see that RF mimicks ZNE well. This result demonstrates that ML-QEM methods can scalably accelerate traditional quantum error mitigation methods by mimicking their behavior when exact expectation values cannot be computed classically. In this experiment, although 1D TFIM is analytically solvable, the Trotter errors should be taken into consid-

eration, and thus the exact expectation values of the circuits are not easily accessible, and thus not shown.

Importantly, this mimicry approach requires less quantum computational overhead *both overall and at runtime*. For this experiment, we test on 40 different coupling strengths J for $h = 0.66\pi$, each of which is used to generate 10 circuits with up to 10 Trotter steps, or 400 test circuits in total. The traditional ZNE approach with 2 noise factors requires $2 \times 400 = 800$ circuits. In contrast, the RF-mimicking-ZNE approach here is trained with 10 different coupling strengths J for $h = 0.66\pi$, each of which generates 10 circuits with up to 10 Trotter steps, or 100 total training circuits. Therefore, the RF-mimicking-ZNE approach requires only $2 \times 100 + 400 = 600$ total circuits, resulting in 25% *overall* lower quantum computational resources. The savings are even more drastic *at runtime*—again, the ZNE approach with 2 noise factors requires 2 circuits to be executed per test circuit, whereas each test circuit only has to be executed once for RF-mimicking-ZNE-based mitigation, resulting in 50% savings. We expect the error of the mimicry to shrink should more training data be provided. We report approximately 0.14 QPU hours (based on a conservative sampling rate of 2 kHz [70]) to generate all the training data and seconds to train the model with a single-core laptop for this experiment.

Efficient Adaptability to Drifted Noise

Because the noise in quantum hardware can drift over time, an ML-QEM model trained on circuits run on a device at one point in time may not perform well at another point in time and may require adaption to the drifted noise model on the device. Therefore, we explore whether an ML-QEM model can be fine-tuned for a different noise model and show that similar performance can be achieved with substantially less training data.

In particular, we fine-tune an MLP and compare its learning rate against RF. The MLP can be fine-tuned on a different noise model after they have been originally trained on a noise model. The fine-tuning is expected to require only a small number of additional samples—this is demonstrated in Fig. 2.8 with the MLP trained on noise model A (`FakeLima`) and fine-tuned on noise model B (`FakeBelem`) which converges after ~ 300 fine-tuning circuits. On the other hand, an MLP trained from scratch and tested on a noise model B shows a slower convergence after ~ 500 training circuits, though both fine-tuning and training from scratch produce the same testing performance. We also compare them with an RF trained from scratch, which converges after fewer than ~ 300 training circuits, demonstrating excellent efficiency in training an RF model. While future research can investigate in more detail the drift in noise affecting the ML model performance, we show evidence that MLP can be efficiently adapted to new device noise and that RF can be trained as efficiently from scratch to new device noise.

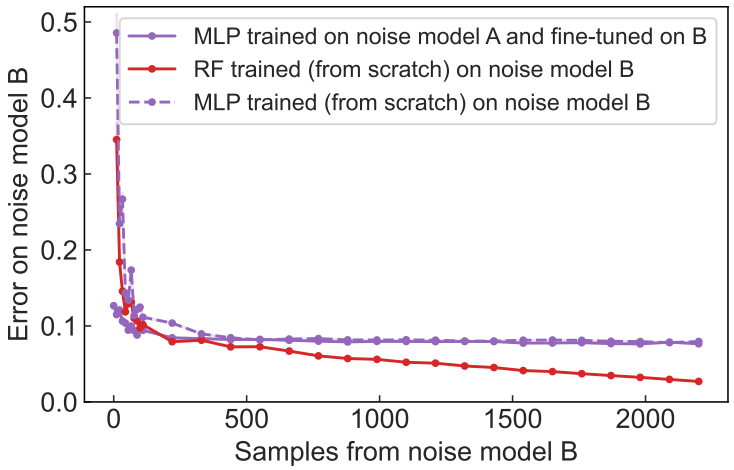


Figure 2.8: Updating the ML-QEM models on the fly. Comparing the efficiency and performance of ML models, fine-tuned or trained from scratch, on a different noise model. Noise model A represents `FakeLima` and noise model B represents `FakeBelem`. All training, fine-tuning, and testing circuits are 4-qubit 1D TFIM measured in a random Pauli basis for four weight-one observables. The solid purple curve shows the testing error on noise model B of an MLP model originally trained on 2,200 circuits run on noise model A and fine-tuned incrementally with circuits run on noise model B. The dashed purple curve shows the testing error on noise model B of another MLP model trained only on circuits from noise model B. The red curve shows the testing error on noise model B of an RF model trained only on circuits from noise model B. All three methods converge with a small number of training/fine-tuning samples from noise model B. While the testing error of the fine-tuned and trained-from-scratch MLP models converged, both were outperformed by a trained-from-scratch RF model. This provides evidence that ML-QEM can be efficient in training.

2.4 Statistical Learning Models

Here, we discuss each of the statistical model (schematics shown in Fig. 2.9), data encoding strategies, and hyperparameters used in this study. We emphasize that the performance of a model depends on factors such as the size of the training dataset, encoding scheme, model architecture, hyper-parameters, and particular tasks. Therefore, from a broader perspective, we hope that the models in this work provide a sufficient starting point for practitioners of quantum computation with noisy devices to make informed decisions about the most suitable approach for their application.

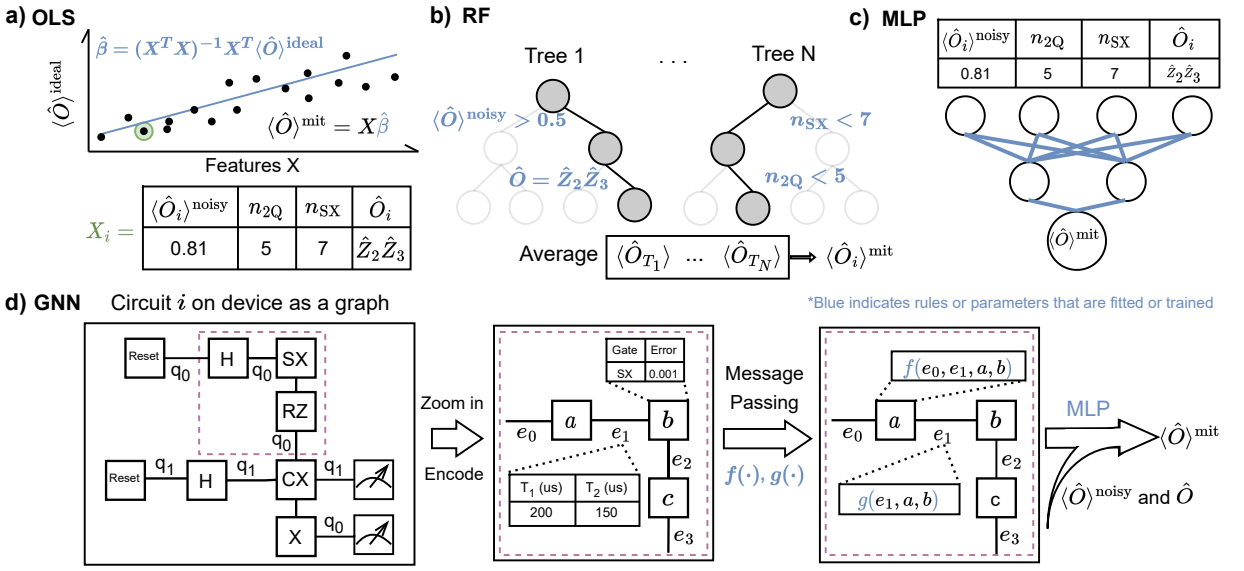


Figure 2.9: Overview of the four ML-QEM models and their encoded features. (a) Linear regression (specifically ordinary least-square (OLS)): input features are vectors including circuit features (such as the number of two-qubit gates n_{2Q} and SX gates n_{SX}), noisy expectation values $\langle\hat{O}\rangle^{\text{noisy}}$, and observables $\hat{O}$. The model consists of a linear function that maps input features to mitigated values $\langle\hat{O}\rangle^{\text{mit}}$. (b) Random forest (RF): the model consists of an ensemble of decision trees and produces a prediction by averaging the predictions from each tree. (c) Multi-layer perception (MLP): the same encoding as that for linear regression is used, and the model consists of one or more fully connected layers of neurons. The non-linear activation functions enable the approximation of non-linear relationships. (d) Graph neural network (GNN): graph-structured input data is used, with node and edge features encoding quantum circuit and noise information. The model consists of multiple layers of message-passing operations, capturing both local and global information within the graph and enabling intricate relationships to be modeled.

Linear Regression

Linear regression is a simple and interpretable method for ML-QEM, where the relationship between dependent variables (the ideal expectation values) and independent variables (the features extracted from quantum circuits and the noisy expectation values) is modeled using a linear function.

One relevant work in this area is Clifford data regression, proposed by Czarnik et al. [83]. In their approach, the authors first replace most of the non-Clifford gates with nearby Clifford gates in the target circuit of interest, then use a linear regression model to regress the noisy expectation values of those circuits onto the ideal ones. Our linear regression model differs

in two main aspects. Firstly, we extend the feature set to include counts of each native gate where native parameterized gates are counted in binned angles, the Pauli observable in sparse Pauli operator representation, and optional device-specific noise parameters. Secondly, our model does not necessarily require training on Clifford versions of the target circuits, although this option remains available if desired.

We train a linear regression model that takes these features as input and predicts the ideal expectation values. The model minimizes the sum squared error between the mitigated and the ideal expectation values using a closed-form solution, which is named ordinary least squares (OLS). The linear regression model can also be trained using other methods, such as ridge regression, LASSO, or elastic net. These methods differ in their regularization techniques, which can help prevent overfitting and improve model generalization. In our experiments, we use OLS for its simplicity and ease of interpretation. We note that standard feature selection procedures also help to prevent overfitting and collinearity in practice.

Random Forest

Random forest (RF) is a robust, interpretable, non-linear decision tree-based model to perform quantum error mitigation. As an ensemble learning method, it employs bootstrap aggregating to combine the results produced from many decision trees, which enhances prediction accuracy and mitigates overfitting. Moreover, each decision tree within the random forest utilizes a random subset of features to minimize correlation between trees, further improving prediction accuracy.

The input features to the random forest model are extracted from the quantum circuits, specifically counts of each native gate on the backend (native parameterized gates are counted in binned angles), the Pauli observable in sparse Pauli operator representation, and optional device-specific noise parameters. We train a random forest regressor with a specified large number of decision trees on the training data. Given all the features, the random forest model averages the predictions from all its decision trees to produce an estimate of the ideal expectation value.

For RF, we used 100 tree estimators for each observable. The tree construction process follows a top-down, recursive, and greedy approach, using the Classification and Regression Trees (CART) algorithm. For the splitting criterion, we employ the mean squared error reduction for regressions. For each tree, at least 2 samples are required to split an internal node, and 1 feature is considered when looking for the best split.

Multi-layer Perceptron

Multi-layer perceptrons (MLPs), first explored in the context of QEM in Ref. [82], are feedforward artificial neural networks composed of layers of nodes, with each layer fully connected to the subsequent one. Nodes within the hidden layers utilize non-linear activation functions, such as the rectified linear unit (ReLU), enabling the MLP to model non-linear relationships.

We construct MLPs with 2 dense layers with a hidden size of 64 and the ReLU activation function. The input features are identical to those employed in the random forest model. To train the MLP, we minimize the mean squared error between the predicted and true ideal expectation values, employing backpropagation to update the neurons. The batch size is 32, and the optimizer used is Adam [55] with an initial learning rate of 0.001. In practice, regularization techniques like dropout or weight decay can be used to prevent overfitting if necessary. The MLP method demonstrates competitive performance in mitigating noisy expectation values, as evidenced by our experiments. However, it should be noted that MLPs are also susceptible to overfitting in this context.

Graph Neural Network

As the most complex model among the four, graph neural networks (GNNs) are designed to work with graph-structured data, such as social networks [98] and chemistry [99]. They can capture both local and global information within a graph, making them highly expressive and capable of modeling intricate relationships. However, their increased complexity results in higher computational costs, and they may be more challenging to implement and interpret.

A core aspect of our ML-QEM with GNN lies in data encoding, which consists of encoding quantum circuits, and device noise parameters into graph structures suitable for GNNs. Before data encoding, each quantum circuit is first transpiled into hardware-native gates that adhere to the quantum device's connectivity. To encode them for GNN, the transpiled circuit is converted into an acyclic graph. In the graph, each edge signifies a qubit that receives instructions when directed towards a node, while each node corresponds to a gate. These nodes are assigned vectors containing information about the gate type, gate errors, as well as the coherence times and readout errors of the qubits on which the gate operates. Additional device and qubit characterizations, such as qubit crosstalk and idling period duration, can be encoded on the edge or node, although they are not considered in the current study.

The acyclic graph of a quantum circuit, serves as input to the transformer convolution layers of the GNN. These message-passing layers iteratively process and aggregate encoded vectors on neighboring nodes and connected edges to update the assigned vector on each node. This enables the exchange of information based on graph connectivity, facilitating the extraction of useful information from the nodes which are the gate sequence in our context. The output, along with the noisy expectation values, is passed through dense layers to generate a graph-level prediction, specifically the mitigated expectation values. As a result, after training the layers using backpropagation to minimize the mean squared error between the noisy and ideal expectation values, the GNN model learns to perform quantum error mitigation.

For the GNN, we use 2 multi-head Transformer convolution layers [100] and ASAPooling layers [101] followed by 2 dense layers with a hidden size of 128. Dropouts are added to various layers. As with the MLP, the batch size is 32, and the optimizer used is Adam [55] with an initial learning rate of 0.001.

2.5 Discussion

In this chapter, we have presented a comprehensive study of machine learning for quantum error mitigation (ML-QEM) methods, including linear regression, random forest, multi-layer perceptrons, and graph neural networks, for improving the accuracy of quantum computations. First, we conducted performance comparisons over many practically relevant contexts; they span circuits (random circuits and Trotterized 1D transverse-field Ising circuits), noise models (qubit decoherence, readout, depolarizing gate, and/or coherent gate errors), and applications (mitigating unseen Pauli observables and enhancing variational quantum eigensolvers) studied here, we find that the best-performing model is the random forest (RF). Notably, RF is a non-linear model that is more complex than linear regression but less so than multi-layer perceptrons and graph neural networks, pointing to the importance of selecting the appropriate model based on the complexity of the target quantum circuit and the desired level of error mitigation. Second, we demonstrated that ML-QEM methods can perform better than a traditional method, zero-noise extrapolation (ZNE). Paired with the ability to mitigate at runtime by running no additional mitigation circuits, ML-QEM reduces the runtime overhead of traditional methods; for instance, it reduces the runtime overhead by a factor of at least 2 compared to digital ZNE. Therefore, ML-QEM can be especially useful for algorithms where many circuits that are similar to each other are executed repeatedly, such as quantum state tomography-like experiments and variational algorithms. Finally, we find that ML-QEM can even effectively mimic other mitigation methods, providing very similar performance but with a lower overhead at runtime. This allows the ML-QEM to scale up to classically intractable circuits.

Future research in ML-QEM can focus on several directions to further enhance the training efficiency, performance, scalability, or generalizability of these methods. First, the training set can be optimized in terms of both the size and type of the training circuits subject to design principles. In the case of the Trotter circuit for example, a simple principle would maximize informational content in training by picking highly different circuits in information content by for example employing symmetry structures in the circuit [84], or focusing on deeper over shallower circuits. Further, one can make assumptions about errors of single-qubit gates [87] or could focus on Clifford or near-Clifford circuits [83, 84, 87, 102]. Second, better encoding strategies that incorporate other significant information about the circuit and the noise model, such as pulse shapes, and output counts, could lead to even more accurate error mitigation. Third, one could study the effect of the drift of noise in hardware on the machine learning model. This would allow one to optimize the resources needed to fine-tune the neural network models or train even to retrain a simple random forest model from scratch periodically. As a step in this direction, we provide evidence that the models can be efficient in fine-tuning to adapt to a change in the noise model, as shown in Sec. 2.3. Fourth, when trained with different noise models and their corresponding noisy expectation values, it is interesting to investigate if setting the noise parameters encoded in GNN, to the low-noise regime (e.g., setting encoded gate error close to zero and the encoded coherence times to a large value) allows the GNN to predict the expectation values close to the ideal

expectation value without ever training on them, and thus providing potential advantages on both accuracy and efficiency when scaling up to classically-intractable circuits. In other words, it is an open question whether the GNN can "extrapolate" to zero noise without specifically amplifying the noise on the target circuit but knowing the noisy expectation values from different noise models on different circuits. Fifth, ML-QEM can be more rigorously optimized and benchmarked against leading methods, such as PEC, PEA, and pulse-stretching ZNE. Finally, extending these methods to other quantum computing tasks and applications will help to further establish the utility of ML-QEM as a powerful tool for improving the accuracy and reliability of quantum computations.

In conclusion, our study underscores the potential of ML-QEM methods for enhancing quantum computations. Understanding the strengths and weaknesses of different models and developing strategies to improve their efficiency paves the way for increasingly robust and accurate applications of quantum computing.

Part II

Quantum Machine Learning

Chapter 3

Adversarial Attacks on Quantum Machine Learning

This chapter is derived from previously published work by Liao, Convy, Huggins, and Whaley [95], which derived the adversarial robustness of general quantum machine learning models with angle encoding classifying general, naturally-generated classical datasets and showed that it only decreases mildly as $\mathcal{O}(1/\sqrt{n})$ in the number of qubits.

3.1 Background on Quantum Machine Learning and Adversarial Attacks

Quantum machine learning (QML) protocols, by exploiting quantum mechanics principles, such as superposition, tunneling, and entanglement [103], have given hope of outperforming their classical counterparts, even with noisy intermediate-scale quantum (NISQ) [104] hardware in the near-term [105]. For classification tasks where statistical patterns can be revealed in complex feature spaces, the high-dimensional Hilbert space of sizable quantum systems offers a naturally advantageous starting ground for QML models. However, many state-of-the-art classical machine learning models, such as deep neural networks with complicated internal feature mappings, have been shown vulnerable to small crafted perturbations to the input, namely adversarial examples [106, 107]. These are intentional worst-case perturbations to the original samples with an imperceptible difference that are nevertheless misclassified by the classifier. This not only raises questions as to why well-performing classifiers suffer from such instabilities but also poses security threats to machine learning applications that emphasize reliability, such as in spam filtering [108]. To understand this unreliable behavior, the transferability of these attacks across different architecture and the robustness against perturbations has led to extensive investigations in the classical machine learning community in recent years [109, 110, 111]. Notably, some geometric and probabilistic arguments, based on curvatures of decision boundaries [112] and the concentration of measure [113, 114, 115, 116, 117], have been employed to quantify the risk of adversarial

attacks in various settings. It has been shown that any classifier will have an adversarial robustness that is increasingly reduced by the dimension of the space on which it classifies, given the concentration of measure phenomenon in certain metric probability spaces [113]. This has raised attention in the QML community where the models take advantage of the high dimensionality of quantum systems [118, 119, 120, 121]

The concentration of measure is a phenomenon that describes the fact that, in certain metric probability spaces, points tend to gather around the boundaries of subsets having at least one half of the probability measure. As a result, there is generically a high probability of obtaining values close to the average for any reasonably smooth function that is evaluated on the distribution [122, 123, 124, 125, 126]. This means that when samples are selected from such a concentrated space, the confidences predicted by the classifier tends to accumulate around the critical value separating the correct and incorrect classes. As such, small targeted perturbations can then easily move the samples across the decision boundary. In particular, it has been recognized that this phenomenon can lead to extreme vulnerabilities of any quantum classifier on high-dimensional Haar-random pure states [118]. Nevertheless, there is no indication of whether such vulnerability exists when classifying on a subset of encoded pure states in a realistic task, such as using a quantum classifier on classical images encoded in pure states.

In this study, we approach the task of classifying quantum states from a geometric perspective. The quantum classifier divides the Hilbert space into subsets, each of which belongs to a certain class. We use this perspective here to study aspects of the problem that are relevant to practical applications of QML. In a practical classification task, such as in recognizing natural images, the samples to be classified can be generated from a Gaussian latent space by one of a number of commonly-used generative models [127, 128, 129, 130, 131]. The success of these models for real-world data generation ensures that the focus on QML models classifying a subset of encoded pure states, where these states are sampled from a distribution that is *smoothly* mapped from a Gaussian latent space [117], will yield insight into the vulnerability of QML models in a real-world classification task. This contrasts with the previous analysis of the vulnerabilities when classifying Haar-random pure states [118].

We demonstrate that the *adversarial robustness* over this generated distribution decreases as $\mathcal{O}(1/\sqrt{n})$ in the number of qubits n, with the scaling measured in the trace norm. This decline in the robustness is mild, indicating a quantum classifier can be robust to attacks on high dimensional quantum states. In contrast, when considering *prediction-change* adversarial settings where the inputs are pure states drawn Haar-randomly, we show that the robustness decreases as $\mathcal{O}(1/2^n)$ in the number of qubits n, implying extreme vulnerabilities to attacks in high-dimensional quantum systems. This second case parallels the result of reference [118], which considered *error-region* adversarial settings and found the robustness also decreases as $\mathcal{O}(1/2^n)$ here. However, we argue that the extreme vulnerability in this setting is not of concern in practice, since the states to be classified are always sampled from a distribution over some subsets of states, rather than from the Haar-random distribution over the entire set of pure states.

The rest of the chapter is structured as follows. In Section 3.1, we introduce the set-

ups and preliminaries in both classical and quantum adversarial attacks. In Section 3.2, we describe the prediction-change adversarial setting, which is often more relevant to real-world classification tasks than the previously employed error-region adversarial setting. We then derive the prediction-change adversarial robustness of any quantum classifier on Haar-randomly distributed pure states and explain its practical limitations. In Section 3.3, we derive the main results on the adversarial robustness of any quantum classifier classifying a smoothly generated distribution over a subset of encoded pure states of interest, and propose a feasible modification to any quantum classifier to lower bound unconstrained adversarial robustness. In Section 3.12, a summary and discussion of the derived robustness over the two types of distribution are presented.

Classical Adversarial Attacks

Classical adversarial attacks were introduced to analyze the instability of deep neural networks caused by a small change to the input sample. Classically, the confidence is often quantified as the probability corresponding to the label class in the output normalized discrete distribution, e.g., the largest softmax value in the output vector in a multi-class logistic-regression convolutional neural network. As numerically shown in various works, such an attack results in a significant drop in the confidence in the correct class [106, 132, 133, 110], and may also bring a significant increase in the confidence in the incorrect class [107]. So far, some arguments have been proposed to explain the vulnerabilities of various classifiers to adversarial attacks and their transferability [107, 134, 135, 136, 116], yet no conclusive consensus has been established [137].

The most common type of adversarial attack is the evasion attack where the adversary does not interfere with the training phase of a classifier and perturbs only the testing samples [109]. The adversary can devise white-box attacks if it possesses total knowledge about the classifier architecture, or otherwise, it can devise black-box attacks relying on the transferability [109, 110]. We shall focus on white-box evasion attacks.

We introduce some notations and definitions used in this study. Let $(\mathcal{X}, \mathrm{d}, \mu)$ denote the sample set $\mathcal{X}$ with a metric d and a probability measure μ. The notation $x \leftarrow \mu$ denotes that a sample x is drawn with a probability measure μ. $\mathcal{L}$ denotes the countable label set. For a subset $\mathcal{S} \subseteq \mathcal{X}$, we let $\mathrm{d}(x, \mathcal{S}) = \inf\{\mathrm{d}(x, y) | y \in \mathcal{S}\}$ and let $B_\epsilon(x) = \{x' | \mathrm{d}(x, x') \leq \epsilon\}$ be the ϵ-neighborhood of x, where d is the metric on $\mathcal{X}$. We also let $\mathcal{S}_\epsilon = \{x | \mathrm{d}(x, \mathcal{S}) \leq \epsilon\}$ be the ϵ-expansion of $\mathcal{S}$. h is a hypothesis or a trained classifier that maps each $x \in \mathcal{X}$ to a predicted label $l \in \mathcal{L}$. c is the ground-truth function that maps each $x \in \mathcal{X}$ to a correct label $l \in \mathcal{L}$. h^l denotes the set of samples classified as label l, namely $h^l = \{x \in \mathcal{X} | h(x) = l\}$. The error region $\mathcal{M}$ is the set of samples on which the hypothesis disagrees with the ground truth, namely $\mathcal{M} = \{x | h(x) \neq c(x)\}$. We define the risk as $R(h, c) = \Pr_{x \leftarrow \mu}[h(x) \neq c(x)] = \mu(\mathcal{M})$.

The two relevant types of evasion attacks studied here are based on the error region and the prediction change. In an error-region attack, the ground-truth function c is accessible and an attack occurs when a perturbation in the sample causes h to disagree with c. In contrast, a prediction-change attack emphasizes the instability of h: an attack occurs when

a perturbation results in a different prediction by h, and c is irrelevant. The precise definitions of these two types of attacks are as follows.

Definition 1 . *The error-region adversarial risk under ϵ-perturbation is the probability of drawing a sample such that its ϵ-neighborhood intersects with the error region,*

$$R_\epsilon^{ER}(h, c, \mu) = \Pr_{x\leftarrow\mu}[\exists x' \in B_\epsilon(x) | h(x') \neq c(x')].$$

Definition 2 . *The prediction-change adversarial risk under ϵ-perturbation is the probability of drawing a sample such that its ϵ-neighborhood contains a sample with a different label,*

$$R_\epsilon^{PC}(h, \mu) = \Pr_{x\leftarrow\mu}[\exists x' \in B_\epsilon(x) | h(x) \neq h(x')],$$

equivalently,

$$R_\epsilon^{PC}(h, \mu) = \Pr_{x\leftarrow\mu}\left[\min_{x'\in\mathcal{X}}\{\mathrm{d}(x', x) | h(x') \neq h(x)\} \leq \epsilon\right].$$

In either type of attack, we call the nearest misclassified examples the adversarial examples. We say that h is more robust if the induced risk of either type is lower for a certain ϵ-perturbation. We shall refer to the minimal ϵ-perturbation to x resulting in an adversarial example as the adversarial perturbation or the robustness of x with h. In contrast, we shall quantify the adversarial robustness of h as the size of ϵ necessary for the adversarial risk of h to be upper bounded by some constant. The main result of this study is an upper bound on the adversarial robustness of any quantum classifier when the input states are smoothly generated from a Gaussian latent space.

Quantum Adversarial Attacks

For our work, a quantum classifier is a quantum channel $\mathcal{E}$ that assigns labels l with some set of positive-operator-valued measures (POVMs) $\{\Pi_l\}$. The quantum classifier takes in an ensemble of identically prepared copies of a state and assigns the state a label l. The confidence of a prediction is quantified as the expectation value of the POVM for the prediction l, namely $\mathrm{tr}(\mathcal{E}(\rho)\Pi_l)$ for an input density matrix ρ. We do not consider the number of copies of a state that is required to implement any specific quantum classification protocol. To measure the perturbation size, the natural choice of metric on quantum states – the trace distance – can be shown to generate an upper bound on the difference between their quantum classification confidence (see Section 3.4), which implies that no small variation can induce a large swing in the predictive confidence. This property of the trace distance is a consequence of its interpretation as the achievable upper bound on the total variation distance[1] between probability distributions arising from measurements performed on those quantum states [8]. Furthermore, we show in Section 3.4 that the Hilbert-Schmidt norm, the Bures distance, and the Hellinger distance between two quantum states all generate a similar upper bound.

[1]Informally, total variation distance is the largest possible difference between the probabilities that the two distributions can assign to the same event.

As a result, in quantum adversarial attacks, the adversary either perturbs the states near the decision boundary minimally to seek misclassification, or aims to maximize confidence change to any state with associated perturbations that are upper bounded by some considerable size in these norms, as illustrated in Figure 3.1. Our work analyzes primarily the risks due to the former objective. In Section 3.5, we also propose a method for the latter objective exploiting the reversibility of parametrized quantum circuits (see e.g. [138, 139]). We note that the latter adversarial setting is justified, since in order to assess the security of a classifier under attack, it is reasonable – given a feasible space of modifications to the input data – to assume that the adversary aims to maximize the classifier's confidence in wrong predictions, rather than merely perturbing minimally in size [110].

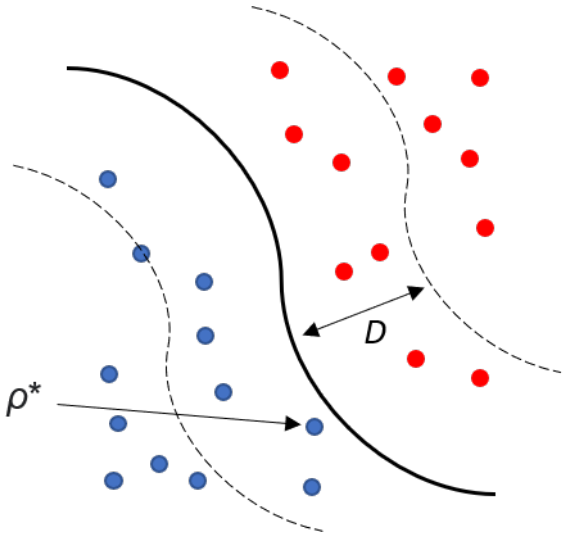


Figure 3.1: The solid curve depicts the decision boundary of a quantum classifier. The states in blue are classified in a different class from the states in red. The metric is the trace distance. The trace distance between any pair of states generates an upper bound on the difference between their quantum classification confidences. Thus ρ^*, the state closest to the decision boundary, is the ideal target of a prediction-change adversarial attack if the adversary aims to achieve misclassifications with minimal perturbations. On the other hand, if the adversary aims to maximize confidence change to any state with associated perturbations of size up to D, then all states between the dashed lines can be perturbed to be misclassified, while all other states can be perturbed to get closer to the boundary, resulting in overall decreased confidence in predicting the correct class. The concentration of measure phenomena implies that for a sufficiently large class, samples tend to lie near the decision boundary.

There are two natural set-ups of adversarial attacks in QML that can be specified. The first is when the input data to the classifier is already quantized and any data transmitted through the quantum communication network comes from an untrusted party. In this case,

the adversary, who may be the sender or an interceptor, can perform an attack either by perturbing each of the transmitted density matrices, or by intercepting a fraction of the copies of the state and substituting them entirely (see Section 3.4). In a broader context, our analysis can be extended to include the instability of classifying quantum states subject to decoherence. We focus on this first set-up in the current study. The second set-up is when the input to the quantum classifier is classical. The quantum classifier encodes the classical data before classifying. Since the adversary is perturbing the classical input data, it is effectively attacking classically. If one views such a quantum classifier as a black-boxed hypothesis function that maps each input to a class, any classifier-agnostic classical analysis of adversarial robustness can then be directly applied. For example, reference [112] analyzes the robustness of any classifier against random or semi-random perturbations, provided the curvature of the decision boundary is sufficiently small, while reference [117] analyzes the adversarial robustness of any classifier when classical input vectors are smoothly mapped from a Gaussian latent representation.

Quantum Data Encoding

We now explain the feature maps used throughout the study. Considering a normalized positive vector u of length n, without loss of generality, we intuitively refer to it as a gray-scale image with n pixels in this study. We focus on a particular set of encoding schemes where the normalized gray-scale value of each pixel, i.e., $u_i \in [0,1], i = 1,\ldots,n$, is featurized into a qubit-encoding state $|\phi_i\rangle$. The product state $|\phi\rangle$ to be classified is a tensor product state of these qubit-encoded pixels in the 2^n-dimensional Hilbert space [140, 141, 142, 143], namely

$$|\phi\rangle = \bigotimes_{i=1}^{n} |\phi_i\rangle = \bigotimes_{i=1}^{n} \left[\cos\left(\frac{\pi}{2}u_i\right)|0\rangle + \sin\left(\frac{\pi}{2}u_i\right)|1\rangle\right]. \tag{3.1}$$

The qubit-encoding states, Eq. (3.1), do not require a quantum random access memory (QRAM) [144] and are efficient in time to prepare. Other schemes including amplitude encoding (see e.g., [145]) are not considered here. We note that some of our results are general and independent of the encoding scheme. We further generalize Eq. (3.1) to qudits. In this case each pixel is mapped to a Hilbert space of higher dimension $d \geq 2$, with the coefficient of the j-th component of the i-th qudit state given by

$$|\phi_i\rangle_j = \sqrt{\binom{d-1}{j-1}} \cos^{d-j}\left(\frac{\pi}{2}u_i\right) \sin^{j-1}\left(\frac{\pi}{2}u_i\right). \tag{3.2}$$

These qudit states are special cases of what are known as spin-coherent states [140], and the qubit states in Eq. (3.1) correspond to $d = 2$.

Concentration of Measure Phenomenon

To describe this phenomenon, let $\Sigma \subseteq \mathcal{X}$ be a Borel set[2]. The concentration function, defined as

$$\alpha(\epsilon) = 1 - \inf_{\Sigma \subseteq \mathcal{X}} \left\{ \mu(\Sigma_\epsilon) \middle| \mu(\Sigma) \geq \frac{1}{2} \right\}, \tag{3.3}$$

has a smaller value when more points are aggregated in the ϵ-expansion of a sufficiently large set Σ, for a fixed ϵ. Informally, a space $\mathcal{X}$ exhibits a concentration of measure if $\alpha(\epsilon)$ decays very fast as ϵ grows, and we shall refer to it as a concentrated space. This is true for a simple example – the standard Gaussian distribution $(\mathbb{R}, \ell^2, \mathcal{N}(0,1))$. Looking at the Borel set $\Sigma = (-\infty, 0)$ whose probability measure is $1/2$, the cumulative density outside its ϵ-expansion, namely $\mathbb{R} \backslash \Sigma_\epsilon = (\epsilon, +\infty)$, decreases at least as fast as $\exp(-\epsilon^2/2)$ by the tail bound [146]. One can invoke isoperimetric inequality [147] to show that this clustering occurs around any Borel set with measure at least $1/2$ and applies to any canonical m-dimensional Gaussian measure in the Euclidean space (see Section 3.10). More formally, a family of N-dimensional spaces with corresponding concentration functions $\alpha_N(\cdot)$ is called a (k_1, k_2)-normal Lévy family if $\alpha_N(\epsilon) \leq k_1 \exp(-k_2^2 \epsilon^2 N)$, where k_1 and k_2 are particular constants. Consequently, the measure is more concentrated for a higher dimension. Two notable normal Lévy families are $\mathbb{SU}(N)$ and $\mathbb{SO}(N)$, both of which are equipped with the Hilbert-Schmidt norm L^2 and the Haar probability measure ν [148, 149]. An implication of this phenomenon is that when points x are drawn from a highly concentrated space, for any function f varying not rapidly, we have $f(x) \approx \langle f \rangle$ with high probability. Lévy's Lemma [122, 123] constitutes a specific example of this.

Related Work

The work in [113] considered any normal Lévy family and derived the robustness for error-region adversarial attacks. The results show that for a nice classification problem[3], if $\mu(\mathcal{M}) = \Omega(1)$, the size of perturbations must be $\mathcal{O}(1/\sqrt{N})$ in order to have the error-region adversarial risk upper bounded by some constant, where N is the dimension of the concentrated space. References [115, 114] studied some specific concentrated spaces and revealed the same scaling.

Reference [118] transforms the classification of pure states $|\phi\rangle$ into that of unitaries U in $|\phi\rangle = U|\vec{0}\rangle$ for some fixed initial state $|\vec{0}\rangle$. These quantum classifiers then classify samples drawn from $\mathbb{SU}(N)$ equipped with the Haar probability measure ν and the Hilbert-Schmidt norm, which is a $(\sqrt{2}, 1/4)$-normal Lévy family. Therefore, if $\mu(\mathcal{M}) > 0$, the necessary condition on the perturbation size for the error-region adversarial risk to be bounded above by $1 - \gamma$ for some $\gamma \in [0,1]$ is $\mathcal{O}(1/\sqrt{N})$. Precisely, to have $R_\epsilon^{ER}(h, c, \nu) \leq 1 - \gamma$, the ϵ-

[2]Borel sets are sets that can be constructed from open or closed sets through countable union, countable intersection, and relative complement

[3]The precise definition of a nice classification problem can be found in Definition 2.3 in [113].

perturbation to any unitary must be upper bounded as[4]

$$\epsilon \leq \sqrt{\frac{4}{N}}\left[\sqrt{\ln\left(\frac{\sqrt{2}}{\mu(\mathcal{M})}\right)}+\sqrt{\ln\left(\frac{\sqrt{2}}{\gamma}\right)}\right]. \tag{3.4}$$

3.2 Problems with Practical Classifications

The result in Eq. (3.4) claims that when classifying unitaries in $\mathbb{SU}(N)$ with the Haar measure, given that an adversary can devise white-box attacks and $\mu(\mathcal{M})$ not exponentially suppressed by N, the robustness of any quantum classifier decreases polynomially in the dimension of the input N. This is daunting since the input has a dimension $N = d^n$ exponential in the number of qudits.

To apply any result related to Eq. (3.4), a ground-truth function c on $\mathbb{SU}(N)$ is needed to obtain the risk $\mu(\mathcal{M})$. However, c may not be easily defined in a real-world machine learning task. For instance, it is challenging to define what constitutes a mistake for visual object recognition. After adding a perturbation to an image, it likely no longer corresponds to a photograph of a real physical scene [150]. Furthermore, it is difficult to define the labels for images undergoing gradual semantic change. All of these factors complicate the evaluation of $\mu(\mathcal{M})$. It thus motivates us to focus on prediction-change adversarial risks (see e.g., [115, 150, 112]) in order to avoid requiring access to the ground truth. The following theorem and corollary then apply.

Theorem 2 . *Let $\mathbb{SU}(N)$ be equipped with the Haar measure ν and the Hilbert-Schmidt norm L^2. For any hypothesis $h : \mathbb{SU}(N) \to \mathcal{L}$ that is not a constant function, let $\eta \in [0, 1/2]$ determine the measure of the dominated class such that $\nu(h^l) \leq 1-\eta, \forall l \in \mathcal{L}$. Suppose $U \in h^l$, $V \notin h^l$ and a perturbation $U \to V$ occurs, where $\|U - V\|_2 \leq \epsilon$. If the prediction-change adversarial risk $R_\epsilon^{PC}(h, \nu) \leq 1-\gamma$, then ϵ must satisfy*

$$\epsilon \leq \sqrt{\frac{4}{N}}\left[\sqrt{\ln\left(\frac{2\sqrt{2}}{\eta}\right)}+\sqrt{\ln\left(\frac{2\sqrt{2}}{\gamma}\right)}\right]. \tag{3.5}$$

It is evident from Eq. (3.5) that the upper bound on the size of the perturbation ϵ is suppressed as the dimension N of the space increases. It is also suppressed when the measure of the dominated class $(1-\eta)$ decreases and when the tolerance on the adversarial risk $(1-\gamma)$ decreases.

Corollary 1 . *With $\rho = U|\vec{0}\rangle\langle\vec{0}|U^\dagger$ and $\sigma = V|\vec{0}\rangle\langle\vec{0}|V^\dagger$, Eq. (3.5) translates to a necessary upper bound in the trace norm between the pure-state density matrices*

$$\|\rho - \sigma\|_1 \leq \frac{4}{N}\lambda_1 = \Omega(d^{-n}).$$

[4] A concise proof of Eq. (3.4) can be found in Section 3.6.

With the qudit encoding in Eq. (3.2), a naive translation of this necessary upper bound to that in the ℓ^1 norm of the encoding vectors u and v gives,

$$\|u - v\|_1 \leq \frac{2n}{\pi} \cos^{-1}\left[\left(1 - \frac{2}{N}\lambda_1\right)^{\frac{1}{(d-1)n}}\right] = \Omega(d^{-\frac{n}{2}}\sqrt{n}),$$

where $N = d^n$ and $\lambda_1 = [\ln(2\sqrt{2}/\eta)]^{1/2} + [\ln(2\sqrt{2}/\gamma)]^{1/2}$ with η and γ defined in Theorem 2.

The proofs can be found in Section 3.7 and 3.8. The interpretation of Theorem 2 and Corollary 1 is clear: given that no class occupies Haar-measure 1, any quantum classifier on quantum states is more vulnerable to prediction-change adversarial attacks on higher-dimensional pure states drawn Haar-randomly, with the robustness decaying exponentially in the number of qudits.

In what follows, we apply this theorem to a practical task by presenting two perspectives on the application, in order to illustrate the limitations of the theorem. Suppose that the objective of the practical task is to classify a subset of quantum states, for example, the pure product states in Section 3.1 that encode images displaying a digit 0 or 1. On one hand, if we label unitaries not related to an actual image, together with unitaries associated with noisy images not displaying a digit 0 or 1, in a third-class, this class will have measure 1, since the set of all unitaries that evolve the initial $|\vec{0}\rangle$ to some final pure product state $|\phi\rangle$ has Haar measure 0 in $\mathbb{SU}(N)$ [151]. For example when $n = 1$, this can be seen by recognizing that the encoded states $\{|\phi\rangle\}$ correspond to only a fraction of the circle going through $|0\rangle$ and $|1\rangle$ on the Bloch Sphere. This labeling renders Theorem 2 useless for any h trained in this way because $\eta = 0$. On the other hand, if we train a binary h to classify half of $\mathbb{SU}(N)$, including unitaries corresponding to 0-digit images, to $l = 0$, and the other half, including unitaries corresponding to 1-digit images, to $l = 1$, then $\eta = 1/2$. Using Eq. (3.5) then gives $\mathcal{O}(1/\sqrt{d^n})$ robustness against prediction-change adversarial attacks, again suggesting extreme vulnerabilities in high dimensions.

However, the interpretation of this result is not of practical interest, for the following reasons. We emphasize that in applying Theorem 2 or Eq. (3.4), the notion of adversarial risks by Definition 2 represents the probability of perturbing a Haar-randomly selected unitary by some ϵ to its adversarial example. It does not represent, for instance, the probability of perturbing a particular unitary associated with a real image to its adversarial example, nor does it represent the risk of attacking a unitary drawn from any other distribution over some subset. Therefore, if the task is to train and generalize a quantum classifier on a subset of quantum states with some distribution, this theorem cannot claim vulnerabilities that are exponential in the number of qudits. It is also noted that, as far as how Eq. (3.4) and Theorem 2 are formulated, the perturbed states cannot be mixed states since these are mapped from $|\vec{0}\rangle\langle\vec{0}|$ by a completely positive and tracing preserving (CPTP) maps rather than by unitaries. In Section 3.3, we shall see that this is an example of an *in-distribution* attack, which applies to scenarios where both the original and perturbed states are pure.

3.3 Robust in Practice: Adversarial Attacks on Quantum Machine Learning

Concentration in Generated Distributions

In practice, one is interested in the performance of a classifier on a distribution over some subset of meaningful samples, such as the subset of images displaying digits including the MNIST data set. It is this distribution on which the adversarial risk should be computed in order to infer the extent of the vulnerability. To ensure that the probability measure on the classifier-input space covers meaningful samples, we resort to approximating the distribution over meaningful samples using the image of a smooth generator function on a concentrated latent space, trained on samples of interest [117]. Following convention, we refer to the latter as a real-data manifold. Such a generator can be a Normalizing Flow model [127, 128] or the generator of a Generative Adversarial Network (GAN) [129, 130, 131], both with a Gaussian latent space, trained on the same data set that the classifier will be trained on. A generative model serving this purpose is also referred to as a spanner [152]. In this way, a major fraction of the samples in the generator output $\mathcal{S}$ can be related to samples of interest, despite the fact that, the smoothness of the generator may introduce some samples off the real-data manifold, such as those undergoing gradual semantic change during interpolations. This generative set-up can be generalized to multiple generators on the same latent space. However, each generator maps to a disjoint part of the real-data manifold, overcoming the problem of covering the off real-data manifold when the latent space is globally connected [153]. This generalization requires relaxing the demand that $\omega(0) = 0$ in the Eq. (3.6) below. As a result, no data off the real-data manifold is generated in $\mathcal{S}$.

The reason that we require the latent space to be concentrated is so that we can study the concentration of samples in the generator-output space resulted from the concentration of the latent space. This connection is made by the assumption that the generator is smooth, in the sense that it admits a modulus of continuity (i.e., it is uniformly continuous), namely if there exists a monotone invertible function $\omega(\cdot)$ such that

$$\|g(z) - g(z')\| \leq \omega(\|z - z'\|_2), \quad \forall z, z' \in \mathcal{Z}, \tag{3.6}$$

where $\|\cdot\|$ is the metric equipped by the image of g. This is a weaker condition than the Lipschitz continuity which results when $\omega(\cdot)$ is a linear function. In this study, we assume $\omega(\cdot)$ to yield a tight upper bound in Eq. (3.6), and we demand $\omega(\tau)$ to be small for small τ for a smooth generator. The idea is that any tendency to concentration of measure in the latent space is preserved by such a smooth mapping to its image, and the generated samples then follow a modified concentrated distribution. We can imagine that if some pairs of latent variables from different classes are within distance b across the class boundary in the generator domain, their generator images must be accordingly within distance at most $\omega(b)$ across the boundary. This can also display a clustering. Although the tendency to cluster is

preserved, the extent to which the points in the generator image gather is mediated by the modulus of continuity. A tight upper bound with $\omega(\cdot)$ that yields distances larger than the typical distances in the output space means that generated samples can be further apart, and vice versa. As far as adversarial robustness is concerned, a larger $\omega(\cdot)$ is then favorable since it implies that larger perturbations are needed to definitively perturb a larger number of generated samples across decision boundaries.

In generating these to-be-classified samples, the fact that a large probability density resides near the decision boundary is not at odds with a trained classifier that predicts training samples with high confidence. The training samples comprise only a subset of the support of the generator-output distribution. High-confidence training samples result from the classifier drawing the decision boundaries away from them. When such a decision boundary encloses a sufficiently large measure, it then inevitably encounters large probability densities – as dictated by the concentration of measure phenomenon on these distributions – that do not contribute to training. As a result, when generalizing to test samples that are similar to the training samples, some test samples may locate near the boundary and be vulnerable targets to adversarial attacks.

Robustness of Quantum Machine Learning Models

We consider the quantum adversarial attack set-up where the input to the classifier is already quantized and transmitted through a quantum communication network.

Let our latent space $\mathcal{Z}$ be, for example in this study, the $\mathbb{R}^m$ with the Euclidean metric ℓ^2 and the canonical m-dimensional Gaussian measure $\mathcal{N}_m \equiv \mathcal{N}(0, I_m)$ so it is a concentrated space. Let $z \leftarrow \mathcal{N}_m$ in $\mathcal{Z}$. Suppose that a smooth generator $g : \mathcal{Z} \to \mathcal{S} \subseteq \mathcal{X}$ is trained to generate a distribution ξ of concern, such as some distribution of natural images, on a subset $\mathcal{S}$ of $\mathcal{X}$. For a sample $g(z) \in \mathcal{S}$, we then have $\xi(g(z)) = \mathcal{N}_m(z)$.

Incorporated in the generator $g = g_2 \circ g_1$, g_1 maps the latent space to a subset of n-pixel natural images, g_2 then encodes the natural image into a density matrix defined in Eq. (3.2). That is, $g(z) = |\phi(z)\rangle\langle\phi(z)| = \rho(z) \in \mathcal{S} \subseteq \mathcal{X}$, where $\mathcal{S}$ – the image of g – is a subset of all density matrices $\mathcal{X}$. The metric on density matrices is the trace norm L^1 unless otherwise specified. The probability measure ξ, which is a distribution mapped by g from the m-dimensional Gaussian measure $\mathcal{N}_m$ on $\mathcal{Z}$, is only supported on $\mathcal{S}$ over density matrices capturing the natural image distribution. Any quantum classifier h then classifies the $d^n \times d^n$ density matrices in $(\mathcal{X}, L^1, \xi)$. Let us denote the intermediate stage – the set of images with n pixels (normalized vectors with length n) – as $\mathcal{I}$, then the corresponding measure on $\mathcal{I}$ can be denoted as $\xi \circ g_2$. The metric on $\mathcal{I}$ is, for instance, the ℓ^1 norm. Diagrammatically, these mappings are

$$\mathcal{Z} \underbrace{\xrightarrow{g_1} \mathcal{I} \xrightarrow{g_2}}_{g} \mathcal{S} \subseteq \mathcal{X} \underset{h}{\longrightarrow} \mathcal{L}.$$

It is noted that smoothness is a desirable property of generative models. It is hinted at gradual transitions in the features in the generated samples, which imply that the generator

has learned relevant factors of variation [154]. We are then justified in assuming that the real-data manifold on $\mathcal{I}$ can be covered by a smooth generator g_1 (see e.g., [128, 129, 130, 131]). In what follows, we show that the overall generator g, mapping from $\mathcal{Z}$ to the real-data manifold in the set of density matrices $\mathcal{X}$, is also smooth.

Proposition 1 . *Assuming that $g_1 : \mathcal{Z} \to \mathcal{I}$ is smooth with a modulus of continuity $\omega_1(\cdot)$ and the qudit encoding scheme, Eq. (3.2), is applied, then the generator $g = g_2 \circ g_1 : \mathcal{Z} \to \mathcal{S} \subseteq \mathcal{X}$ is also smooth and admits a modulus of continuity $\omega(\cdot)$ that is lower bounded as*

$$\omega(\tau) \geq \sqrt{1 - \cos^{2n(d-1)}\left(\frac{\pi}{2n}\omega_1(\tau)\right)}, \quad \forall \tau \geq 0.$$

The proof can be found in Section 3.9. In terms of the scaling with respect to n and d, when $\omega_1(\cdot)$ scales as $\Omega(1)$, for instance, when g_1 is Lipschitz continuous (e.g., the generator in [155, 156]), Proposition 1 implies that the modulus of continuity of the overall generator g, i.e., $\omega(\cdot)$, scales as $\Omega(\sqrt{d/n})$. It is desirable to enforce Lipschitz continuity on some generators, for example when imposing spectral normalization [157] on the generator of a GAN to improve training [156].

A distinction can be made concerning whether the adversarial example σ must be also in the subset $\mathcal{S}$. If so, the adversarial attack is called in-distribution, since the attacker only looks for an adversarial example within the data manifold $\mathcal{S}$. Otherwise, we call it an unconstrained adversarial attack since the perturbation is arbitrary in $\mathcal{X}$, i.e., it is not confined to the data manifold. We state the precise definitions, based on prediction-change adversarial risks in Definition 2, as follows.

Definition 3 . *An in-distribution adversarial attack, or a data-manifold attack, attempts to find the perturbation*

$$\begin{aligned}\varepsilon_{in}(\rho) &= \min_{r \in \mathcal{Z}} \{\|g(z+r) - \rho\|_1 | h(g(z+r)) \neq h(\rho)\} \\ &= \min_{\sigma \in \mathcal{S}} \{\|\sigma - \rho\|_1 | h(\sigma) \neq h(\rho)\},\end{aligned}$$

which is within the data manifold $(\mathcal{S}, L^1, \xi)$. It induces an in-distribution adversarial risk,

$$R^{PC}_{\epsilon_{in}}(h, \xi) = \Pr_{\rho \leftarrow \xi}\left[\varepsilon_{in}(\rho) \leq \epsilon_{in}\right].$$

Definition 4 . *An unconstrained adversarial attack attempts to find*

$$\varepsilon_{unc}(\rho) = \min_{\sigma \in \mathcal{X}} \{\|\sigma - \rho\|_1 | h(\sigma) \neq h(\rho)\},$$

which is in $(\mathcal{X}, L^1)$ not restricted to the data manifold $\mathcal{S}$. It induces an unconstrained adversarial risk,

$$R^{PC}_{\epsilon_{unc}}(h, \xi) \equiv R^{PC}_{\epsilon}(h, \xi) = \Pr_{\rho \leftarrow \xi}\left[\varepsilon_{unc}(\rho) \leq \epsilon\right].$$

It is noted that when the generator is surjective on $\mathcal{X}$, i.e., $\mathcal{S} = \mathcal{X}$, there is no distinction between the two types of attacks. The set-ups in Theorem 2 and Eq. (3.4) consider classifying on the subset of all pure-state density matrices in $\mathcal{X}$ on which a Haar-random distribution ν is supported. Since this requires both the original and perturbed states to be pure, the adversarial risks are considered in-distribution, although we shall see in Section 3.3 that the same upper bound applies to the unconstrained robustness for a general quantum classifier.

In-distribution Adversarial Robustness

The following theorem, depending on the distribution to be classified as well as the specific classical-data generator g_1 in terms of $\omega_1(\cdot)$, then applies.

Theorem 3 . *Let $h : \mathcal{X} \to \mathcal{L}$ be any quantum classifier on the set of density matrices. Considering in-distribution adversarial attacks on the image of g, if $\xi(h^l)) \leq 1/2, \forall l$, i.e., the classes are not too unbalanced, then for the prediction-change risk $R^{PC}_{\epsilon_{in}}(h, \xi) \leq 1 - \gamma$, the distance between two density matrices ϵ_{in} must satisfy*

$$\epsilon_{in} \leq \omega\left(\sqrt{\ln\left(\frac{\pi}{2\gamma^2}\right)}\right), \tag{3.7}$$

where $\omega(\cdot)$ is the modulus of continuity in Proposition 1.

The proof can be found in Section 3.10. This result is independent of the quantum data encoding scheme. It can be generalized to quantum classifiers with arbitrary decision boundaries, but in this case, the necessary upper bound on the in-distribution robustness will not have a closed form (see Section 3.10). This upper bound is saturated when Eq. (3.6) is tight and the quantum classifier induces linearly separable regions in the latent space, namely when $h \circ g$ is a linear function on $\mathcal{Z}$, giving rise to the maximally robust quantum classifier. The non-saturation of this upper bound when class regions are not linearly separable in $\mathcal{Z}$ can be seen in the example of the standard Gaussian in Section 3.1 above. Suppose one looks at $\Sigma' = (-\infty, -2\delta) \cup (0, 2\delta)$ for some $\delta > 0$, which has the same probability measure $1/2$ as $\Sigma = (-\infty, 0)$ but is not linearly separable in $\mathbb{R}$. The measure outside the δ-expansion of Σ', i.e., $\mathbb{R} \setminus \Sigma'_\delta = (3\delta, +\infty)$, is smaller than that outside of the δ-expansion of Σ, namely $\mathbb{R} \setminus \Sigma_\delta = (\delta, +\infty)$, implying more concentration outside and near Σ' than Σ.

The non-saturation of this upper bound for non-linearly separable classification regions in $\mathcal{Z}$ also implies that it is prone to misclassification with an increasing number of equiprobable classes. The proof for cases with at least 5 equiprobable classes can be found in Section 3.10. Informally, more equiprobable classes lead to more boundaries, enclosing classes with sufficiently large total measure, that border distinct classes. Then within a fixed distance beyond more of those boundaries, there are more samples subject to some prediction change.

We note that this upper bound is usually not saturated in practice, since a quantum classifier is usually linear, such as a parametrized quantum circuit and a unitary tensor network, while the generator g is usually non-linear, given that g_1 is usually non-linear and

g_2, the quantum feature map, is non-linear. Classically, some highly-nonlinear state-of-the-art neural networks have robustness one or two orders of magnitude smaller in the ℓ^2 norm on some data sets than the corresponding upper bound derived with similar arguments [117]. It would be interesting to examine the amount of deviation from the upper bound for QML models in future works.

Theorem 3 implies that when the quantum states to be classified encode classical data generated with a modulus of continuity scaling as $\Omega(1)$, the in-distribution robustness of any quantum classifier decreases polynomially in the number of qudits n and increases polynomially in the qudit dimension d. To see this, we first note that according to Proposition 1, when $\omega_1(\cdot) = \Omega(1)$, which applies to generators such as those enforcing Lipschitz continuity, $\omega(\cdot)$ is lower bounded by a function that scales as $\Omega(\sqrt{d/n})$. This means that the upper bound on the perturbation size ϵ_{in} between any two in-distribution states, i.e., the right-hand side of Eq. (3.7), is then asymptotically bounded from below by $\sqrt{d/n}$.

As such, the vulnerability increases slightly with a larger number of qudits n and by contrast, decreases slightly with qudits of higher dimension $d \geq 2$. When the encoded classical data manifold comes from generators for which Lipschitz continuity is not enforced, it requires numerical approximations of the modulus of continuity $\omega_1(\cdot)$ to determine its scaling in the output space, before obtaining the robustness scaling. Compared to Theorem 2 where samples are Haar-random pure states, the states to be classified here, which characterize the adversarial risk, are similar to those considered in practical tasks. Specifically, they are a subset of encoded states with a distribution smoothly generated from a Gaussian latent space. Theorem 3 demonstrated that, contrary to previous claims [118], there is no guarantee that quantum classifiers are exponentially more vulnerable to in-distribution attacks in higher-dimensional Hilbert space. We shall now show that the theorem applies to unconstrained attacks as well.

Unconstrained Adversarial Robustness

Unconstrained adversarial attacks are arbitrary perturbations in $\mathcal{X}$ to a sample ρ. In a broader context in which the instability of the quantum classifier is concerned, this may derive from density matrices subject to decoherence in a classification task. It is clear that $\varepsilon_{unc}(\rho) \leq \varepsilon_{in}(\rho), \forall \rho \in \mathcal{X}$ and thus, it holds by changing the in-distribution perturbations in Theorem 3 to unconstrained ones, and the same bound in Eq. (3.7) applies.

We argue that there does not exist a tighter upper bound that holds for general quantum classifiers for unconstrained robustness. Consider a particular family of quantum classifiers that project any state onto the data manifold, namely to map any state to its closest in-distribution state, before classifying. These classifiers can be shown to satisfy $1/2\varepsilon_{in}(\rho) \leq \varepsilon_{unc}(\rho) \leq \varepsilon_{in}(\rho), \forall \rho \in \mathcal{X}$[5]. Even in the worst case where $\varepsilon_{unc}(\rho) = 1/2\varepsilon_{in}(\rho), \forall \rho \in \mathcal{X}$, their unconstrained robustness is as large as half of the in-distribution one. We stress that,

[5]It is proven in Theorem 2 in [117].

although robust, such a quantum classifier is inefficient in our setting since there is no apparent tractable way to obtain the closest pure product state to an arbitrary state.

$\leq 1-\gamma$	$\|\rho-\sigma\|_1 \leq$	$\|u-v\|_1 \leq$
$R_\epsilon^{PC}(h,\nu)$	$4d^{-n}\lambda_1 = \Omega(d^{-n})$	$\frac{2n}{\pi}\cos^{-1}\left[(1-2d^{-n}\lambda_1)^{\frac{1}{(d-1)n}}\right]$
$R_\epsilon^{PC}(h,\xi)$	$\omega(\lambda_2) \geq \sqrt{1-\cos^{2n(d-1)}\left(\frac{\pi}{2n}\omega_1(\lambda_2)\right)} = \Omega\left(\sqrt{\frac{d}{n}}\right)$	$\omega_1(\lambda_2) = \Omega(1)$

Table 3.1: Summary of the adversarial robustness, namely the size of perturbations necessary for the adversarial risk to be upper bounded by some constant, of any quantum classifier obtained within the prediction-change adversarial attack setting. In this setting, the prediction-change adversarial risk over the Haar-random distribution ν ($R_\epsilon^{PC}(h,\nu)$) and over a smoothly generated distribution ξ ($R_\epsilon^{PC}(h,\xi)$) are both upper bounded by $(1-\gamma)$ (column 0). d denotes the qudit dimension in Eq. (3.2) and n denotes the number of encoded qudits or the length of the encoding vectors (number of pixels in the image classification example). Parameters λ_1 and λ_2 are defined as $\lambda_1 = [\ln(2\sqrt{2}/\eta)]^{1/2} + [\ln(2\sqrt{2}/\gamma)]^{1/2}$ and $\lambda_2 = \sqrt{\ln(\pi/(2\gamma^2))}$. Row 1 summarizes the adversarial robustness when a pure state ρ sampled from the Haar-random distribution ν is perturbed to a state σ. The robustness is shown both in the trace norm (column 1), as well as in its translation to the robustness measured in the ℓ^1 norm of the set of encoding vectors (from Corollary 1 of Theorem 2) (column 2). Both upper bounds decrease exponentially in n. Row 2 summarizes the adversarial robustness when a pure state ρ sampled from a smoothly generated distribution ξ from a Gaussian latent space is perturbed to a state σ (column 1), and the robustness when the intermediately generated vector u is perturbed to v (column 2) (from Proposition 1 and Theorem 3) . Note that when the robustness in adversarially perturbing a vector scales as $\Omega(1)$, e.g., when the intermediate vectors are generated Lipschitz continuously, that in perturbing an encoded pure state scales as $\Omega(\sqrt{d/n})$.

Inspired by this strategy, we propose that one can construct a family of efficient quantum classifiers $\tilde{h}$ on n-qubit density matrices $\mathcal{X}$ with unconstrained robustness $\varepsilon_{unc}(\rho)$ lower bounded for any $\rho \in \mathcal{X}$. To be specific, we construct $\tilde{h}$ from any h with the following procedure.

Let the original sample $\rho \in \mathcal{S}$ be a pure product-state density matrix with n qudits as in Eq. (3.1). A perturbation $\epsilon_{unc} \equiv \epsilon$ leads to a sample $\sigma \in \mathcal{X}$. First, we perform single qubit tomography on every qubit of σ to reconstruct a product-state density matrix from these single qubits. We denote this mapping as $P : \mathcal{X} \to \mathcal{X}$, $\sigma \mapsto \bigotimes_{i=1}^n \mathrm{tr}_{\{j\neq i\}}(\sigma)$. Subsequently, we numerically fit the pixel values $\{s_i\}$ to $P(\sigma)$ to find its closest density matrix $\tilde{\sigma}$ within our data manifold $\mathcal{S}$. We use a symbol $\tilde{\sigma}$ to represent the density matrix attained from this

procedure. $\tilde{\sigma}$ is then replacing σ when fed to the quantum classifier h. We have the following theorem,

Theorem 4 . *For every n-qubit $\rho \in \mathcal{S} \subseteq \mathcal{X}$, let $\tilde{\rho}$ be the density matrix within the data manifold attained from the above procedure. For any quantum classifier h, let $\tilde{h} : \mathcal{X} \to \mathcal{L}$ be such that $\tilde{h}(\rho) = h(\tilde{\rho})$, then*

$$2 - 2\left(1 - \frac{\varepsilon_{in}(\rho)^2}{16}\right)^{\frac{1}{n_e}} \leq \varepsilon_{unc}(\rho) \leq \varepsilon_{in}(\rho),$$

where $n_e = n$ for even n and $n_e = n + 1$ for odd n.

The proof can be found in Section 3.11. We note that the procedure can be applied to any product state encoding scheme. This procedure yields an explicit lower bound to the unconstrained adversarial perturbation when it is possible to estimate the in-distribution adversarial perturbation by, for example, sampling in the latent space [158] or gradient descent search in the latent space [152] before mapping to the density matrices. This $\tilde{h}$ constructed from h amounts to a feasible tomographic preprocessing of input states. It guarantees that the unconstrained robustness of each sample ρ is bounded from below and may be used as a defense strategy against unconstrained adversarial attacks in practice. However, we note that when n is large, this lower bound can be several orders of magnitude smaller than the upper bound.

3.4 Confidence Difference and Distance between States

We show that the predictive confidence difference in any QML protocol is upper bounded by the distance between the input density matrices up to some constant factor, where this distance is measured in the trace norm L^1, the Hilbert-Schmidt norm L^2, the Bures distance, or the Hellinger distance.

Considering density matrices ρ and σ, the trace norm between them is defined to be $\|\rho - \sigma\|_1 = \mathrm{tr}(|\rho - \sigma|)$. Consider a set of POVMs $\{\Pi_l\}$ and a quantum channel $\mathcal{E}$ such that $\mathcal{E}(\rho) = \sum_i M_i \rho M_i^\dagger$ and $\sum_i M_i^\dagger M_i = I$. We have,

$$\begin{aligned}
\mathrm{tr}(\mathcal{E}(\rho)\Pi_l) - \mathrm{tr}(\mathcal{E}(\sigma)\Pi_l) &= \mathrm{tr}\left(\sum_i M_i(\rho - \sigma)M_i^\dagger \Pi_l\right) \\
&= \mathrm{tr}\left((\rho - \sigma)\sum_i M_i^\dagger \Pi_l M_i\right) \\
&\equiv \mathrm{tr}((\rho - \sigma)\mathcal{E}^*(\Pi_l)).
\end{aligned}$$

We note that $\mathcal{E}^*$ is the dual map of $\mathcal{E}$ and $\{\mathcal{E}^*(\Pi_l)\}$ is still a set of POVMs, since $\mathcal{E}^*(\Pi_l)$ is hermitian, non-negative because $\mathrm{tr}(\rho\mathcal{E}^*(\Pi_l)) = \mathrm{tr}(\mathcal{E}(\rho)\Pi_l) \geq 0$, and complete because $\sum_{i,s} M_i^\dagger \Pi_l M_i = \sum_i M_i^\dagger M_i = I$.

For each particular measurement, we can expand in its eigenbasis $\mathcal{E}^*(\Pi_l) = \sum_k b_k |\phi_k\rangle\langle\phi_k| \equiv \sum_k b_k P_k$. Let $\{|\psi_i\rangle\}$ and $\{\lambda_i\}$ be the eigenbasis and eigenvalues of $(\rho - \sigma)$, so $\|\rho - \sigma\|_1 =$

$\sum_i |\lambda_i| \in [0,2]$. We then expand $\mathcal{E}^*(\Pi_l) = \sum_{i,j,k} b_k a_{ik} |\psi_i\rangle a^*_{jk} \langle\psi_j|$ such that $\sum_i |a_{ik}|^2 = 1, \forall k$ and $\sum_k b_k = \mathrm{tr}(\mathcal{E}^*(\Pi_l)) \geq 0$ due to the non-negativity. We have

$$\begin{aligned}
\mathrm{tr}((\rho-\sigma)\mathcal{E}^*(\Pi_l)) &= \mathrm{tr}\left((\rho-\sigma)\sum_{i,j,k} b_k a_{ik} |\psi_i\rangle a^*_{jk} \langle\psi_j|\right) \\
&= \sum_k b_k \,\mathrm{tr}\left(\sum_{i,j} a_{ik} a^*_{jk} \langle\psi_j| (\rho-\sigma) |\psi_i\rangle\right) \\
&= \sum_{i,k} b_k |a_{ik}|^2 \lambda_i \leq \sum_k b_k \|\rho-\sigma\|_1 \\
&= \mathrm{tr}(\mathcal{E}^*(\Pi_l))\|\rho-\sigma\|_1.
\end{aligned} \tag{3.8}$$

Therefore,

$$|\mathrm{tr}(\mathcal{E}(\rho)\Pi_l) - \mathrm{tr}(\mathcal{E}(\sigma)\Pi_l)| \leq \mathrm{tr}(\mathcal{E}^*(\Pi_l))\|\rho-\sigma\|_1.$$

When $\mathrm{tr}(\mathcal{E}^*(\Pi_l))$ is not too large the above inequality suggests that the confidence change will be small when the trace norm between the two density matrices is small. However, $\mathrm{tr}(\mathcal{E}^*(\Pi_l))$ may be very large in high dimensions and in that case, the upper bound becomes very weak.

We resort instead to the physical interpretation of trace distance being a generalization of the classical total variation distance. The trace distance between two quantum states is an achievable upper bound on the total variation distance between probability distributions arising from measurements performed on those states [8]:

$$\frac{1}{2}\|\rho-\sigma\|_1 = \frac{1}{2}\max_{\{\Pi_l\}} \sum_l |\mathrm{tr}[(\rho-\sigma)\Pi_l]|,$$

where the maximization is over all POVMs $\{\Pi_l\}$ and the factor of 2 is to restrict the maximal trace distance to be 1. Using the contractive property of the trace norm under any CPTP map, we conclude that the trace norm constitutes an upper bound to the sum of confidence changes of all POVMs:

$$\sum_l |\mathrm{tr}(\mathcal{E}(\rho-\sigma)\Pi_l)| \leq \|\mathcal{E}(\rho) - \mathcal{E}(\sigma)\|_1 \leq \|\rho-\sigma\|_1. \tag{3.9}$$

Considering the Hilbert-Schmidt norm defined as $\|\rho-\sigma\|_2^2 = \mathrm{tr}[(\rho-\sigma)^2]$, if we regard $\|\rho-\sigma\|_2$ as the inner product of the two vectors $(1,1,\cdots,1)$ and $(|\lambda_0|,|\lambda_1|,\cdots,|\lambda_{N-1}|)$, then from the Cauchy-Schwarz inequality we find $\|\rho-\sigma\|_1 \leq \sqrt{N}\|\rho-\sigma\|_2$. But this bound is very weak in high-dimensional Hilbert space. A better upper bound is given in [159] that $\|\rho-\sigma\|_1 \leq 2\sqrt{R}\|\rho-\sigma\|_2$, where $R = \mathrm{rank}(\rho)\mathrm{rank}(\sigma)/[\mathrm{rank}(\rho)+\mathrm{rank}(\sigma)]$. This implies that, even when one state is full rank, if the other state is low rank, then the Hilbert-Schmidt norm is of the same order of magnitude as the trace norm. This is the case when we consider any perturbation to an encoded pure state density matrix ρ whose rank is 1. Combined with Eq. (3.9), we arrive at a similar upper bound,

$$\sum_l |\mathrm{tr}(\mathcal{E}(\rho)\Pi_l) - \mathrm{tr}(\mathcal{E}(\sigma)\Pi_l)| \leq 2\sqrt{R}\|\rho-\sigma\|_2.$$

Considering the Bures distance defined as $\|\rho-\sigma\|_B^2 = 2(1-\sqrt{F(\rho,\sigma)})$, it is an extension to mixed states of the Fubini-Study distance for pure states [160]. We have

$$\|\rho-\sigma\|_1 \leq 2\sqrt{1-\left(1-\frac{1}{2}\|\rho-\sigma\|_B^2\right)^2} = 2\sqrt{\|\rho-\sigma\|_B^2-\frac{1}{4}\|\rho-\sigma\|_B^4} \leq 2\|\rho-\sigma\|_B,$$

where the first inequality is proven in [161, 160] and saturated for pure states. Therefore, together with Eq. (3.9), we conclude that

$$\sum_l |\mathrm{tr}(\mathcal{E}(\rho)\Pi_l) - \mathrm{tr}(\mathcal{E}(\sigma)\Pi_l)| \leq 2\|\rho-\sigma\|_B. \tag{3.10}$$

Finally, considering the Hellinger distance defined as $\|\rho-\sigma\|_H^2 = 2-2\,\mathrm{tr}(\sqrt{\rho}\sqrt{\sigma})$, it is shown that $\|\rho-\sigma\|_B \leq \|\rho-\sigma\|_H$ [160] and thus, the same upper bound applies by changing $\|\rho-\sigma\|_B$ to $\|\rho-\sigma\|_H$ in Eq. (3.10).

In QML, if ρ and σ are close in these norms and are separated by any class boundary, say between class $l=s$ and class $l=t$, then $\mathrm{tr}(\mathcal{E}(\rho)\Pi_s) > \mathrm{tr}(\mathcal{E}(\sigma)\Pi_s)$ while $\mathrm{tr}(\mathcal{E}(\rho)\Pi_t) < \mathrm{tr}(\mathcal{E}(\sigma)\Pi_t)$. This suggests that no small perturbation to density matrices in these norms can significantly change the measurement outcome and thus, alter the prediction, unless the original sample is near the boundary. In other words, viewing $\mathrm{tr}(\mathcal{E}(\rho)\Pi_s)$ as the confidence of predicting $l=s$, it implies that no small perturbations can result in a high-confidence sample in one class perturbed to a low-confidence sample in the same class, or a high-confidence sample in a different class.

3.5 Adversarial Attacks Exploiting Quantum Classifier Reversibility

We propose a method to perform adversarial attacks in our first set-up in Section 3.1 on quantized data. This method can be carried out on quantum hardware when the computation is classically intractable. We assume a noiseless QML model for this analysis, so the quantum channel is unitary. Considering, for example, the unitary tree tensor network (TTN) in [138, 162] among other types of parametrized unitary quantum circuits, the adversary can run it reversely starting from a density matrix with any designated wrong class label $l=t$ such that $\mathrm{tr}(\sigma'\Pi_t)=1$ while $\mathrm{tr}(\sigma'\Pi_{l\neq t})=0$. Any qubit that is traced out in the forward direction is initialized to an arbitrary state and passes through the network in the reverse direction. The output of the reversal circuit is a set of density matrices $\{U^\dagger\sigma' U\} \equiv \{\sigma\}$ such that $\mathrm{tr}(U\sigma U^\dagger\Pi_t)=1$ whereas $\mathrm{tr}(U\sigma U^\dagger\Pi_{l\neq t})=0$. Thus, this set of density matrices will be classified in the wrong class by the POVM Π_t with high confidence. Suppose that the original samples are $\{\rho\}$ in the class $s\neq t$ and $\mathrm{tr}(U\rho U^\dagger\Pi_s) = 1/2+\delta$ with some $\delta\in(0,1/2]$. The adversary then replaces an ϵ-portion of the transmitted quantum states $\{\rho\}$ with the $\{\sigma\}$ to attack the receiver.

To achieve a prediction change, the adversary demands $\mathrm{tr}(U[(1-\epsilon)\rho+\epsilon\sigma]U^\dagger\Pi_s) < 1/2$. This requires

$$\epsilon > 1 - \frac{1}{1+2\delta}, \tag{3.11}$$

which means that the portion of $\{\rho\}$ being substituted with $\{\sigma\}$ increases with higher-confidence of $\{\rho\}$. We note that this effectively creates a perturbation of size

$$\begin{aligned}\|\rho - [(1-\epsilon)\rho+\epsilon\sigma]\|_1 &\geq \epsilon \sum_l |\mathrm{tr}(U(\rho-\sigma)U^\dagger\Pi_l)| \\ &= \epsilon\Big[\sum_{l\neq t}\mathrm{tr}(U\rho U^\dagger\Pi_l) + (1-\mathrm{tr}(U\rho U^\dagger\Pi_t))\Big] \\ &= \epsilon[2-2\,\mathrm{tr}(U\rho U^\dagger\Pi_t)] \geq \epsilon(1+2\delta),\end{aligned}$$

where the first inequality follows from Eq. (3.9). As a result, a misclassification by the attack demands a perturbation of size $\|\rho-[(1-\epsilon)\rho+\epsilon\sigma]\|_1 \geq 2\delta$ where we substituted in Eq. (3.11).

3.6 Proof of Eq. (3.4)

We present a condensed proof based on the proof to Theorem 3.7 in [113]. Let $\epsilon_1 > \sqrt{1/(Nk_2)\ln(k_1/\mu(\mathcal{M}))}$ and $\epsilon_2 > \sqrt{1/(Nk_2)\ln(k_1/\gamma)}$. Then the concentration function satisfies $\alpha(\epsilon_1) < \mu(\mathcal{M})$ and $\alpha(\epsilon_2) < \gamma$. As such, by directly applying Part 2 of Theorem 3.2 in [113], we conclude $R_\epsilon^{ER}(h,c,\nu) > 1-\gamma$ for $\epsilon = \epsilon_1+\epsilon_2$. It can be shown that $\mathbb{SU}(N)$ is a $(\sqrt{2},1/4)$-normal Lévy family and so $k_1 = \sqrt{2}$ and $k_2 = 1/4$ [118]. The contrapositive statement on $R_\epsilon^{ER}(h,c,\nu) \leq 1-\gamma$ then gives the necessary condition Eq. (3.4).

3.7 Proof of Theorem 2

Proof. We let $\epsilon_1 > \sqrt{1/(Nk_2)\ln(2k_1/\eta)}$ and $\epsilon_2 > \sqrt{1/(Nk_2)\ln(2k_1/\gamma)}$, then the concentration function satisfies $\alpha(\epsilon_1) < \eta/2$ and $\alpha(\epsilon_2) < \gamma/2$. Therefore, by applying Part 1 of Theorem A.2 in [113], we conclude that for $\epsilon = \epsilon_1+\epsilon_2$, $R_\epsilon^{PC}(h,\nu) > 1-\gamma$. For completeness, we present our explained version of the proof below.

Let $\epsilon = \epsilon_1+\epsilon_2$. By assumption that $\nu(h^l) \leq 1-\eta, \forall l \in \mathcal{L}$, it can be easily verified by contradiction that $\exists l_1 \in \mathcal{L}$ s.t. $\nu(h^{l_1}) \in (\eta/2, 1/2]$. Let $h^{l_1,C} = \mathcal{X}\setminus h^{l_1}$. On one hand, we know that $\nu(h^{l_1}) > \eta/2 > \alpha(\epsilon_1)$ where the last inequality is given by our assumption. We prove by contradiction that $\nu(h_{\epsilon_1}^{l_1}) > 1/2$. Suppose not, then we have for $\mathcal{S} = \mathcal{X}\setminus h_{\epsilon_1}^{l_1}$, $\nu(\mathcal{S}) = 1-\nu(h_{\epsilon_1}^{l_1}) \geq 1/2$. Then by the definition of the concentration function in Eq. (3.3), $\nu(\mathcal{S}_{\epsilon_1}) \geq 1-\alpha(\epsilon_1)$. Combining with what we obtained that $\nu(h^{l_1}) > \alpha(\epsilon_1)$, we have $\nu(\mathcal{S}_{\epsilon_1}) + \nu(h^{l_1}) > 1$. Thus, $\exists x \in \nu(\mathcal{S}_{\epsilon_1}) \cup \nu(h^{l_1})$. This implies $\exists y \in \mathcal{S} | \mathrm{d}(y,x) \leq \epsilon_1$. But this same y must also be in $h_{\epsilon_1}^{l_1}$ since the same x is also in h^{l_1}. However, this raises a contradiction since $\mathcal{S}$ and $h_{\epsilon_1}^{l_1}$ are disjoint by definition, i.e., $\nexists y | y \in \mathcal{S}, y \in h_{\epsilon_1}^{l_1}$. Now, $\nu(h_{\epsilon_1}^{l_1}) > 1/2$ means, by the definition of

the concentration function in Eq. (3.3), as well as the assumption that $\gamma/2 > \alpha(\epsilon_2)$, we have $\nu(h_\epsilon^{l_1}) \geq 1 - \alpha(\epsilon_2) > 1 - \gamma/2$.

On the other hand, knowing that $\nu(h^{l_1,C}) \geq 1/2$, we have that $\nu(h_{\epsilon_2}^{l_1,C}) > 1 - \gamma/2$ followed by simply replacing the $h_{\epsilon_1}^{l_1}$ in the previous sentence with $h^{l_1,C}$ since they both have measure at least 1/2. We then also have $\nu(h_\epsilon^{l_1,C}) > 1 - \gamma/2$. Hence, using the inequality $\mu(\cap_{i=1}^n A_i) \geq \sum_{i=1}^n \mu(A_i) - (n-1)$, one can conclude that $\nu(h_\epsilon^{l_1} \cap h_\epsilon^{l_1,C}) > 1 - \gamma$ and so, by the prediction-change risk's definition, $R_\epsilon^{PC}(h,\nu) \geq \nu(h_\epsilon^{l_1} \cap h_\epsilon^{l_1,C}) > 1 - \gamma$.

It can be shown that $\mathbb{SU}(N)$ is a $(\sqrt{2}, 1/4)$-normal Lévy family and so $k_1 = \sqrt{2}$ and $k_2 = 1/4$ [118]. The contrapositive statement on $R_\epsilon^{PC}(h,\nu) \leq 1 - \gamma$ then gives the necessary condition Eq. (3.5). □

3.8 Proof of Corollary 1

Proof. We have from Theorem 2 that the necessary condition for $R_\epsilon^{PC}(h,\nu) \leq 1 - \gamma$ on $\mathbb{SU}(N)$ is $\|U - V\|_2 \leq \sqrt{4/N}\lambda_1$ where $\lambda_1 = \left[[\ln(2\sqrt{2}/\eta)]^{1/2} + [\ln(2\sqrt{2}/\gamma)]^{1/2}\right]$. Let $\sigma = V|\vec{0}\rangle\langle\vec{0}|V^\dagger$. From the Proof of Theorem 3 in [118], we have $\|U - V\|_2^2 \geq 2N(1 - |\langle\phi|\psi\rangle|)$. The Fuchs–van de Graaf inequality for pure states is

$$2 - 2\sqrt{F(\rho,\sigma)} \leq \|\rho - \sigma\|_1 = 2\sqrt{1 - F(\rho,\sigma)}, \tag{3.12}$$

where the fidelity $F(\rho,\sigma) = |\langle\phi|\psi\rangle|^2$. Based on Eq. (3.12), we obtain

$$2N(1 - |\langle\phi|\psi\rangle|) \geq \frac{2NT(\rho,\sigma)^2}{(1 + |\langle\phi|\psi\rangle|)} \geq NT(\rho,\sigma)^2,$$

where T is the trace distance. As such, we need

$$\sqrt{\frac{4}{N}}\lambda_1 \geq \|U - V\|_2 \geq \sqrt{N}T(\rho,\sigma) = \frac{\sqrt{N}}{2}\|\rho - \sigma\|_1,$$

which gives $\|\rho - \sigma\|_1 \leq 4/N\lambda_1 = 4d^{-n}\lambda_1$.

We translate this upper bound on the distance between two density matrices to that between their encoding vectors $g_1(z)$ and $g_1(z')$. Altogether with the necessary condition and Eq. (3.12), we have

$$4d^{-n}\lambda_1 \geq \|\rho - \sigma\|_1 \geq 2 - 2\sqrt{F(\rho,\sigma)}. \tag{3.13}$$

For density matrices $\rho, \sigma \in \mathcal{X}$ respective to two images, we have $\rho = |\phi\rangle\langle\phi| = \bigotimes_i |\phi_i\rangle\bigotimes_i\langle\phi_i| = \bigotimes_i |\phi_i\rangle\langle\phi_i| = \bigotimes_i \rho_i$ and $\sigma = \bigotimes_i |\psi_i\rangle\langle\psi_i| = \bigotimes_i \sigma_i$, which are mapped from images $g_1(z) = \vec{s}$ and $g_1(z') = \vec{t}$, respectively. All i-indices run from 1 to n. And $|\phi_i\rangle$ and $|\psi_i\rangle$ are featurized from pixels of value s_i and t_i, respectively. It can be shown by induction that

$$F(\rho,\sigma) = \prod_i \cos^{2(d-1)}\left(|s_i - t_i|\frac{\pi}{2}\right). \tag{3.14}$$

For $d = 2$, we have that $F(\rho, \sigma) = \mathrm{tr}(\bigotimes_i \rho_i \bigotimes_i \sigma_i) = \prod_i \mathrm{tr}(\rho_i \sigma_i) = \prod_i |\langle \phi_i | \psi_i \rangle|^2 = \prod_i \cos^2(|s_i - t_i|\pi/2)$. It then suffices to show $\langle \phi_i | \psi_i \rangle = \cos^{d-1}(|s_i - t_i|\pi/2)$ for the qudit encoding $d > 2$. We drop all $\pi/2$ factors and the subscripts i in s_i and t_i hereafter. Suppose for $d = k$, we have $\langle \phi_i | \psi_i \rangle$ equal to

$$\sum_{j=1}^{k} \binom{k-1}{j-1} \cos^{k-j}(s) \cos^{k-j}(t) \sin^{j-1}(s) \sin^{j-1}(t) = \cos^{k-1}(s-t). \tag{3.15}$$

Then for $d = k + 1$, we have $\langle \phi_i | \psi_i \rangle$ equal to

$$\begin{aligned}
&\sum_{j=1}^{k+1} \binom{k}{j-1} \cos^{k+1-j}(s) \cos^{k+1-j}(t) \sin^{j-1}(s) \sin^{j-1}(t) \\
&= \cos(s)\cos(t) \left[\sum_{j=1}^{k} \beta \binom{k}{j-1} \cos^{k-j}(s) \cos^{k-j}(t) \sin^{j-1}(s) \sin^{j-1}(t) \right] \\
&\quad + \sin(s)\sin(t) \left[\sum_{j=2}^{k+1} (1-\beta) \binom{k}{j-1} \cos^{k+1-j}(s) \cos^{k+1-j}(t) \sin^{j-2}(s) \sin^{j-2}(t) \right],
\end{aligned} \tag{3.16}$$

where $\beta = (k + 1 - j)/k$.

Identifying the two expressions in the square brackets as both equal to Eq. (3.15), we obtain the desired outcome $\langle \phi_i | \psi_i \rangle = \cos^k(s - t)$, and the induction completes.

Combining Eq. (3.13) and Eq. (3.14), we have

$$4d^{-n}\lambda_1 \geq 2 - 2\prod_i \cos^{d-1}\left(|s_i - t_i|\frac{\pi}{2}\right) \geq 2 - 2\cos^{(d-1)n}\left(\frac{\sum_i |s_i - t_i|}{n}\frac{\pi}{2}\right). \tag{3.17}$$

where the last inequality follows from the inequality $\cos^n(\sum_i x_i/n) \geq \prod_i \cos(x_i)$. It can be readily shown for $n \geq 2$ using the following trick. Consider any pair x_i and x_j and let x_m be their arithmetic average so $x_i = x_m + d$ and $x_j = x_m - d$ for some $d \neq 0$. Then $\cos(x_i)\cos(x_j) = \cos(x_m + d)\cos(x_m - d) = \cos^2(x_m) - \sin^2(d) \leq \cos^2(x_m)$. Therefore, one can maximize the overall cosine product, while maintaining the sum of the arguments, by replacing any pair $\cos(x_i)$ and $\cos(x_j)$ with $\cos(x_m)$ and $\cos(x_m)$, and successively replacing every pair till every factor converges to $\cos(\sum_i x_i/n)$ with the same argument.

Solving for $\sum_i |s_i - t_i| = \|g_1(z) - g_1(z')\|_1$ in Eq. (3.17) yields the upper bound on the perturbation size in $(\mathcal{I}, \ell^1)$. □

3.9 Proof of Proposition 1

Proof. We decompose g into $g_2 \circ g_1$ where $g_1 : (\mathcal{Z}, \ell^2) \to (\mathcal{I}, \ell^1)$ is desired to be smooth in practice. It can be generalized to ℓ^p norm on $\mathcal{I}$ and similar proof follows since the ℓ^p norm of any given vector does not grow with p. We have $\|g_1(z) - g_1(z')\|_1 \leq \omega_1(\|z - z'\|_2), \forall z, z' \in \mathcal{Z}$.

We show that it is also smooth for the qudit encoding $g_2 : (\mathcal{I}, \ell^1) \to (\mathcal{X}, L^1)$ as in Eq. (3.2). Applying the qudit feature map and similar to that in Section 3.8, it can be shown that

$$\|\rho - \sigma\|_1 = 2\sqrt{1 - \prod_i \cos^{2(d-1)}\left(|s_i - t_i|\frac{\pi}{2}\right)}. \tag{3.18}$$

Since $\omega(\cdot)$ is used in an upper bound in Theorem 3, we need to obtain the scaling of a lower bound to $\omega(\cdot)$. The $\omega(\cdot)$ that forms a tight upper bound in Eq. (3.6) must have $\omega(\|z - z'\|_2)$ upper bounding Eq. (3.18) for arbitrary $z, z' \in \mathcal{Z}$. Hence, it is equivalent to finding the scaling of a lower bound to Eq. (3.18). That is, we have $\forall z, z' \in \mathcal{Z}$,

$$\begin{aligned}
\omega(\|z - z'\|_2) &\geq 2\sqrt{1 - \prod_i \cos^{2(d-1)}\left(|s_i - t_i|\frac{\pi}{2}\right)} \\
&\geq 2\sqrt{1 - \cos^{2(d-1)n}\left(\frac{\sum_i |s_i - t_i|}{n}\frac{\pi}{2}\right)} \\
&= 2\sqrt{1 - \cos^{2(d-1)n}\left(\frac{\pi}{2n}\|g_1(z) - g_1(z')\|_1\right)},
\end{aligned}$$

where the second inequality follows from the inequality $\cos^n(\sum_i x_i/n) \geq \prod_i \cos(x_i)$ proven for Eq. (3.17). Since the above inequality holds for any z, z' such that $\|z - z'\|_2 = \tau$ for any τ, and since we assume $\omega(\cdot)$ forms a tight upper bound in Eq. (3.6), g is smooth with

$$\omega(\tau) \geq \sqrt{1 - \cos^{2n(d-1)}\left(\frac{\pi}{2n}\omega_1(\tau)\right)}, \quad \forall \tau > 0.$$

In terms of the scaling with respect to n and d, if $\omega_1(\cdot) = \Omega(1)$, such as when g_1 is Lipschitz continuous, we have $\omega(\cdot) = \Omega(\sqrt{d/n})$. □

3.10 Proof of Theorem 3

Proof. If letting $\epsilon_{in} \geq \omega(\sqrt{\ln[\pi/(2\gamma^2)]})$, then $\gamma \geq \sqrt{\pi/2}\exp(-\omega^{-1}(\epsilon_{in})^2/2)$. By the definition of the generator and the latent space, we have $\mathcal{N}_m(g^{-1}(\rho)) = \xi(\rho)$, $\forall \rho \in \mathcal{S} \subseteq \mathcal{X}$. Let us define $h^i_{\to} = \{\rho \in h^i | \mathrm{d}(\rho, \cup_{j\neq i} h^j) \leq \epsilon_{in}\}$ which is the set of density matrices that are at positive distance at most ϵ_{in} from $\cup_{j\neq i} h^j$, then following Definition 3,

$$\begin{aligned}
R^{PC}_{\epsilon_{in}}(h, \xi) &= \Pr_{\rho \leftarrow \xi}[\min_{\sigma \in \mathcal{S}}\{\|\sigma - \rho\|_1 | h(\sigma) \neq h(\rho)\} \leq \epsilon_{in}] \\
&= \xi(\cup_i h^i_{\to})) = \mathcal{N}_m(g^{-1}(\cup_i h^i_{\to})),
\end{aligned} \tag{3.19}$$

since $h^i_{\to}$ are disjoint for different class i. Hence, it can be shown that $R^{PC}_{\epsilon_{in}}(h, \xi) \geq 1 - \gamma$ when $\xi(h^i) \leq 1/2, \forall i$ from Theorem 1 in [117]. The contrapositive yields the necessary condition Eq. (3.7). For completeness, we present our condensed version of the proof below.

We write the cumulative distribution function of the standard Gaussian distribution $\mathcal{N}(0, 1)$ as $\Phi(x) = 1/\sqrt{2\pi}\int_{-\infty}^{x}\exp(-u^2/2)\mathrm{d}u$.

Theorem 5 (Gaussian isoperimetric inequality)[147, 122]. *Let $\mathcal{N}_m$ be the canonical Gaussian measure on $\mathbb{R}^m$. Let $\Sigma \subseteq \mathbb{R}^m$ be any Borel set and let $\Sigma_\epsilon = \{z \in \mathbb{R}^m | \exists z' \in \Sigma \text{ s.t. } \|z - z'\|_2 \leq \epsilon\}$. If $\mathcal{N}_m(\Sigma) = \Phi(a)$ then $\mathcal{N}_m(\Sigma_\epsilon) \geq \Phi(a + \epsilon)$.*

Lemma 1 [117]. *Let $p \in [1/2, 1]$, we have for all $\eta > 0$,*

$$\Phi(\Phi^{-1}(p) + \eta) \geq 1 - (1-p)\sqrt{\frac{\pi}{2}} e^{-\frac{\eta^2}{2}}. \tag{3.20}$$

If $p = 1 - 1/K$ for $K \geq 5$ and $\eta \geq 1$, we have

$$\Phi(\Phi^{-1}(1 - \frac{1}{K}) + \eta) \geq 1 - \frac{1}{K}\sqrt{\frac{\pi}{2}} e^{-\frac{\eta^2}{2}} e^{-\eta\sqrt{\log\left(\frac{K^2}{4\pi \log(K)}\right)}}. \tag{3.21}$$

We first introduce the following sets in the latent space $(\mathbb{R}^m, \ell^2, \mathcal{N}_m)$: $H^i = g^{-1}(h^i)$ and $H^i_{\to} = \{z \in H^i | \mathrm{d}(z, \cup_{j\neq i} H^j) \leq \omega^{-1}(\epsilon_{in})\}$. We note that $H^i_{\to} \bigcup \cup_{j\neq i} H^j$ is the set of points that are at distance at most $\omega^{-1}(\epsilon_{in})$ from $\cup_{j\neq i} H^j$. Then by Theorem 5 applied with $\Sigma = \cup_{j\neq i} H^j$ and $a = a_{\neq i} \equiv \Phi^{-1}(\mathcal{N}_m(\cup_{j\neq i} H^j))$, we have $\mathcal{N}_m(H^i_{\to}) + \mathcal{N}_m(\cup_{j\neq i} H^j) \geq \Phi(a_{\neq i} + \omega^{-1}(\epsilon_{in}))$. Rearranging, $\mathcal{N}_m(H^i_{\to}) \geq \Phi(a_{\neq i} + \omega^{-1}(\epsilon_{in})) - \Phi(a_{\neq i})$. As $H^i_{\to}$ are disjoint for different class i, we have

$$\mathcal{N}_m(\cup_i H^i_{\to}) \geq \sum_{i=1}^{K} \left[\Phi(a_{\neq i} + \omega^{-1}(\epsilon_{in})) - \Phi(a_{\neq i})\right].$$

By the definition of $\omega(\cdot)$, we have $g(H^i_{\to}) \subseteq h^i_{\to}$. It leads to $\mathcal{N}_m(g^{-1}(h^i_{\to})) \geq \mathcal{N}_m(H^i_{\to})$ and $\mathcal{N}_m(\cup_i g^{-1}(h^i_{\to})) \geq \mathcal{N}_m(\cup_i H^i_{\to})$. Therefore, we obtain the result for arbitrary decision boundary,

$$\mathcal{N}_m(\cup_i g^{-1}(h^i_{\to})) \geq \sum_{i=1}^{K} \left[\Phi(a_{\neq i} + \omega^{-1}(\epsilon_{in})) - \Phi(a_{\neq i})\right].$$

Equivalently by Eq. (3.19),

$$R^{PC}_{\epsilon_{in}}(h, \xi) \geq \sum_{i=1}^{K} \left[\Phi(a_{\neq i} + \omega^{-1}(\epsilon_{in})) - \Phi(a_{\neq i})\right].$$

Suppose $\xi(h^i) = \mathcal{N}_m(H^i) \leq 1/2$ and $\mathcal{N}_m(\cup_{j\neq i} H^j) \geq 1/2, \forall i$. Using Eq. (3.20) in Lemma 1 in the second inequality below,

$$\begin{aligned}
R^{PC}_{\epsilon_{in}}(h, \xi) &\geq \sum_{i=1}^{K} \left[\Phi(\Phi^{-1}(\mathcal{N}_m(\cup_{j\neq i} H^j)) + \omega^{-1}(\epsilon_{in})) - \mathcal{N}_m(\cup_{j\neq i} H^j)\right] \\
&\geq \sum_{i=1}^{K} \left[1 - (1 - \mathcal{N}_m(\cup_{j\neq i} H^j))\sqrt{\frac{\pi}{2}} e^{\frac{-\omega^{-1}(\epsilon_{in})^2}{2}} - \mathcal{N}_m(\cup_{j\neq i} H^i)\right] \\
&= \left(1 - \sqrt{\frac{\pi}{2}} e^{\frac{-\omega^{-1}(\epsilon_{in})^2}{2}}\right) \sum_{i=1}^{K} \left[1 - \mathcal{N}_m(\cup_{j\neq i} H^i)\right] \\
&= 1 - \sqrt{\frac{\pi}{2}} e^{\frac{-\omega^{-1}(\epsilon_{in})^2}{2}} > 1 - \gamma,
\end{aligned}$$

provided that $\gamma > \sqrt{\pi/2}\exp(-\omega^{-1}(\epsilon_{in})^2/2)$. The contrapositive yields the results in our Theorem 3 that $\epsilon_{in} \leq \omega(\sqrt{\ln[\pi/(2\gamma^2)]})$ is necessary for $R^{PC}_{\epsilon_{in}}(h,\xi) \leq 1-\gamma$.

When there are at least 5 equiprobable classes [117], substituting Eq. (3.21) in Lemma 1 into the above inequality yields

$$R^{PC}_{\epsilon_{in}}(h,\xi) \geq 1 - \sqrt{\frac{\pi}{2}} e^{\frac{-\omega^{-1}(\epsilon_{in})^2}{2}} e^{-\epsilon_{in}\sqrt{\log\left(\frac{K^2}{4\pi\log(K)}\right)}}.$$

Hence, the in-distribution robustness of h decreases with the number of equiprobable classes.

Alternatively, a numerically looser upper bound on ϵ_{in} can be derived from the fact that $(\mathbb{R}^m, \ell^2, \mathcal{N}_m)$ resembles a normal Lévy family but the concentration function decays independently of N. By Theorem 5, any Borel set Σ there such that $\mathcal{N}_m(\Sigma) = \Phi(a)$ satisfies $\mathcal{N}_m(\Sigma_\epsilon) \geq \Phi(a+\epsilon)$. In particular, for all Borel sets A with measure at least $1/2$, we have $a \geq 0$ and thus, $1 - \mathcal{N}_m(A_\epsilon) \leq 1 - \Phi(\epsilon) \leq \exp(-\epsilon^2/2)$ where the last inequality follows from the Gaussian tail bound. By definition of the concentration function in Eq. (3.3), $\alpha(\epsilon) = \sup_A\{1 - \mathcal{N}_m(A_\epsilon)\} \leq \exp(-\epsilon^2/2)$.

By substituting the statement and the proof of Theorem 2 with $k_1 = 1$ and $k_2 = 1/\sqrt{2}$ and $N = 1$, we have the following. Let $\eta \in [0, 1/2]$ be such that $\mathcal{N}_m(H^l) = \xi(h^l) \leq 1-\eta,\ \forall l \in \mathcal{L}$. If $\epsilon_{in} \geq \omega(\sqrt{\ln(4/\gamma^2)} + \sqrt{\ln(4/\eta^2)})$, then by acting $\omega^{-1}(\cdot)$, which is a strictly increasing function, on both sides, we obtain $\omega^{-1}(\epsilon_{in}) \geq \sqrt{\ln(4/\gamma^2)} + \sqrt{\ln(4/\eta^2)}$. This implies that $R^{PC}_{\omega^{-1}(\epsilon_{in})}(h, \mathcal{N}_m) \geq 1-\gamma$. Since $R^{PC}_{\omega^{-1}(\epsilon_{in})}(h, \mathcal{N}_m) \leq R^{PC}_{\epsilon_{in}}(h,\xi)$ (this is equivalent to $g(H^i_{\rightarrow}) \subseteq h^i_{\rightarrow}$), it therefore implies $R^{PC}_{\epsilon_{in}}(h,\xi) \geq 1-\gamma$. The contrapositive yields, for $R^{PC}_{\epsilon_{in}}(h,\xi) \leq 1-\gamma$, it is necessary to have $\epsilon_{in} \leq \omega(\sqrt{\ln(4/\gamma^2)} + \sqrt{\ln(4/\eta^2)})$. When $\eta = 1/2$, it can be verified that this necessary upper bound is looser than that in Theorem 3 for the same γ.

□

3.11 Proof of Theorem 4

Proof. We have the mapping to obtain a product state density matrix $P : \mathcal{X} \to \mathcal{X},\ \sigma \mapsto \bigotimes_{i=1}^n \mathrm{tr}_{\{j\neq i\}}\sigma$ where n is the number of qubits. This is not a CPTP map on the set of $d^n \times d^n$ density matrices $\mathcal{X}$ since it is non-linear. Nonetheless, it can be viewed as a CPTP map Λ on $\mathcal{X}^{\otimes n}$ as $\Lambda : \mathcal{X}^{\otimes n} \to \mathcal{X},\ \sigma^{\otimes n} \mapsto \mathrm{tr}_{\{j\neq i\}}([\sigma^{\otimes n}]_i)$ where $[\sigma^{\otimes n}]_i$ denotes the i-th copy of σ, which involves only partial tracing. In particular, for a product state $\rho^{\otimes a}$ with the integer $a \geq 1$, $\Lambda(\rho^{\otimes a}) = \rho$.

Consider $\rho \in \mathcal{S} \subseteq \mathcal{X}$ an n-qubit density matrix, namely $\rho = g(z)$ for some $z \in \mathcal{Z}$. Let $\sigma \in \mathcal{X}$. We have

$$\begin{aligned}\|\rho - P(\sigma)\|_1 = \|\Lambda(\rho^{\otimes n}) - \Lambda(\sigma^{\otimes n})\|_1 &\leq \|\rho^{\otimes n} - \sigma^{\otimes n}\|_1 \\ &\leq 2\sqrt{1 - F(\rho^{\otimes n}, \sigma^{\otimes n})} = 2\sqrt{1 - F(\rho,\sigma)^n},\end{aligned}$$

where the first inequality follows from the contractive property of the trace norm under any CPTP map and the last equality follows from the multiplicativity of fidelity with respect

to tensor products. By Eq. (3.12), we have $F(\rho,\sigma) \geq (1 - \|\rho - \sigma\|_1/2)^2$. Substituting in, we obtain

$$\|\rho - P(\sigma)\|_1 \leq 2\sqrt{1 - \left(1 - \frac{\|\rho - \sigma\|_1}{2}\right)^{2n}}.$$

Let $\tilde{\sigma} \in \mathcal{S}$ be the closest in-distribution sample to $P(\sigma)$, which can be found by fitting parameters $\{s_i\}$ in Eq. (3.1). Therefore, $\|P(\sigma) - \tilde{\sigma}\|_1 \leq \|P(\sigma) - \rho\|$. We then obtain

$$\|\rho - \tilde{\sigma}\|_1 \leq \|\rho - P(\sigma)\|_1 + \|P(\sigma) - \tilde{\sigma}\|_1 \leq 4\sqrt{1 - \left(1 - \frac{\|\rho - \sigma\|_1}{2}\right)^{2n}}. \tag{3.22}$$

Recall that for the quantum classifier $\tilde{h}$, $\tilde{h}(\sigma) = h(\tilde{\sigma})$. Taking minimum over all σ such that $\tilde{h}(\sigma) \neq \tilde{h}(\rho)$ (i.e., $h(\tilde{\sigma}) \neq h(\rho)$),

$$\varepsilon_{in}(\rho) \leq \min\{\|\rho - \tilde{\sigma}\|_1\} \leq 4\sqrt{1 - \left(1 - \frac{\min\{\|\rho - \sigma\|_1\}}{2}\right)^{2n}}, \tag{3.23}$$

we obtain

$$\varepsilon_{in}(\rho) \leq 4\sqrt{1 - \left(1 - \frac{\varepsilon_{unc}(\rho)}{2}\right)^{2n}}. \tag{3.24}$$

Notice that to obtain an inequality between $\varepsilon_{in}(\rho)$ and $\varepsilon_{unc}(\rho)$ like in Eq. (3.24), it is sufficient to have Eq. (3.23) hold after taking the minimum, and it is not necessary to have Eq. (3.22) hold for a generic σ. Since for n-qubit density matrices which are separable with respect to some equal bipartition of the system, denoted as $\{\rho_b\}$, form a dense subset [163], we can effectively realize the same minimum in Eq. (3.23) over $\sigma \in \{\rho_b\}$ such that $\tilde{h}(\sigma) \neq \tilde{h}(\rho)$ instead. For equal bipartite states, the number of copies to make a CPTP map Λ' acting on them to obtain $P(\sigma)$ reduces to $n/2$ if n is even and reduces to $(n+1)/2$ if n is odd. For instance, given a 4-qubit σ whose qubit 1 is only entangled with 2 and qubit 3 is only entangled with 4, $\Lambda'(\sigma^{\otimes 2}) = \mathrm{tr}_{\{1,3\}}(\sigma) \otimes \mathrm{tr}_{\{2,4\}}(\sigma) = P(\sigma) = \Lambda(\rho^{\otimes 4})$. Therefore, we can replace the exponent $1/(2n)$ in Eq. (3.24) with $1/n$ for even n and $1/(n+1)$ for odd n.

We recall $\varepsilon_{unc}(\rho) \leq \varepsilon_{in}(\rho),\ \forall \rho \in \mathcal{X}$ and rearrange,

$$2 - 2\left(1 - \frac{\varepsilon_{in}(\rho)^2}{16}\right)^{\frac{1}{n_e}} \leq \varepsilon_{unc}(\rho) \leq \varepsilon_{in}(\rho),$$

where $n_e = n$ for even n and $n_e = n + 1$ for odd n. □

3.12 Discussion

A summary of the upper bounds on the prediction-change adversarial robustness over pure states sampled from the Haar-random distribution ν and a smoothly generated distribution ξ, is presented in Table 3.1.

In this work, we first showed the prediction-change adversarial robustness over Haar-randomly distributed pure states, and compared this with the previously demonstrated error-region robustness of [118] over the same distribution. Both types of adversarial robustness show similar extreme vulnerabilities exponential in the number of qudits. However, in this work, we have argued that these vulnerabilities for Haar-random pure states are not of practical interest. This is because, in practice, the adversarial risk of a quantum classifier should be computed on a distribution over some subset of meaningful states, such as a subset of qubit encoding states featurizing some images, in order to infer the extent of the vulnerability. In general, practical quantum classification tasks classify a subset of encoded states with some commonly used qubit encoding schemes. For such tasks, we have shown that we can use the concentration of measure phenomenon to derive the robustness of any quantum classifiers in situations where the distribution of states to be classified can be smoothly generated from a Gaussian latent space, as quantified in Eq. (3.6). In this situation, we have shown that one finds only a mildly polynomially decreasing robustness in the number of such encoded qubits, specifically with scaling as $\mathcal{O}(\sqrt{1/n})$ in the trace norm.

As noted for Theorem 3, it is the upper bound on the perturbation size necessary for the adversarial risk to be bounded from above that scales as $\Omega(\sqrt{1/n})$. This upper bound is usually not tight and the actual adversarial robustness could therefore be smaller. We have also proposed a feasible modification of any quantum classifier with product-state inputs – namely, by performing single qubit tomography before numerically fitting the closest encoded qubit state – to obtain a lower bound on the unconstrained robustness and to defend against unconstrained adversarial attacks.

Most importantly, our analysis provides QML protocols some relief from adversarial attacks in real-world tasks. For example, when classifying on some qubit states encoding MNIST images, the robustness decreases only as $\mathcal{O}(\sqrt{1/n})$, in contrast to the extreme vulnerability of quantum classifiers in classifying Haar-random pure states (Theorem 2 and [118]). In the future, it will be interesting to experimentally compare the adversarial robustness of particular QML models for real-world data on a distribution of states smoothly mapped from a Gaussian latent space with the bounds that we have derived here.

We note that the polynomially decreasing robustness in n is derived from the qudit encoding scheme. The concentration of measure due to the Gaussian isoperimetric inequality for the latent space only contributes to the argument of Eq. (3.7). It will be interesting to investigate whether a different encoding scheme can give better scaling in the robustness, and also to determine whether quantum data that derives naturally from a distribution other than the Haar-random distribution is robust to attacks. In Section 3.5, we propose a method to perform white-box adversarial attacks on classically intractable input states with QML models. It will be interesting to further explore white-box attacks, assuming that the adversary is capable of devising these. In practice, with current NISQ-era hardware, it will also be useful to examine how robust QML models are against adversarial attacks under noise and decoherence.

Chapter 4

Tensor Network Quantum Machine Learning Models

This chapter is derived from the previously published work by Liao, Convy, Yang, and Whaley [162], which analyzed and simulated the competing effect between having decoherence in the tensor network quantum machine learning models and increasing bond dimension of the network. Liao contributed primarily to all of the numerical experiments, and both Liao and Ian contributed to the theoretical analysis.

4.1 Background on Tensor Networks

Tensor networks (TNs) are compact data structures engineered to efficiently approximate certain classes of quantum states used in the study of quantum many-body systems. We often encounter high-dimensional objects in many-body quantum physics—when dealing with the large number of component spaces, we suffer the "curse of dimensionality". As a simple example, a general quantum state of a lattice of spin-$\frac{1}{2}$ fermions can be written as $|\psi\rangle = \sum_{i_1,i_2,\ldots,i_n} C_{i_1 i_2 \ldots i_n} |i_1\rangle |i_2\rangle \cdots |i_n\rangle$. It is difficult to store and manipulate the complex-valued high-order tensor $C_{i_1 i_2 \ldots i_n}$ given the exponentially scaling number of elements (2^n) specifying it. However, should there be a way to form this high-order tensor from smaller component tensors through contraction, e.g., $C_{i_1 i_2 \ldots i_n} = \sum_{j,k=1}^{2} A_{i_1 \ldots i_{\frac{n}{2}} jk} B_{jk i_{\frac{n}{2}+1} \ldots i_n}$, we only need to store two lower order tensors A and B, which have a total size of $2 \times 2^{\frac{n}{2}+2} \ll 2^n$ as n becomes large. This illustrates the principle of tensor network as a compact representation of wavefunctions. Such a contraction of lower-order tensor can be not exact—the matrix product state (MPS) with a small bond dimension often requires the truncation of small singular values in its construction through successive singular value decomposition (SVD). To simplify the computation and presentation of these lower-order component tensors, a set of diagrammatic rules consistent with linear algebra is used to indicate the contraction of different dimensions, leading to the commonly-used Penrose graphical notation, or tensor diagrams. We provide one example of such on copy tensors as follows.

Copy Tensors

A copy tensor of order n is defined to be $\Delta_n = \sum_i e_i^{\otimes n}$ where e_i is the ith basis vector, whose conventional tensor diagram is given as a solid dot with n bonds [164, 165]. An order-one copy tensor contraction can be viewed as a marginalization, while an order-three copy tensor can be used to denote conditioning on the same vector, as shown in Fig. 4.1. The contraction of two third-order copy tensors with a density matrix and with themselves while leaving two bonds uncontracted is taking the diagonals of a matrix: $\delta_{ikl} M_{ij} \delta_{jkm} = M_{ij} \delta_{ijlm} = M_{ij} \delta_{ij}$.

Figure 4.1: Left: using a third-order copy tensor contracting with a basis state vector results in an outer product of the basis vector, which can be thought of as conditioning on the same basis state upon contraction with two nodes. Right: Obtaining the diagonals of a density matrix, or a matrix in general, can be done by contracting the matrix with two third-order copy tensors and contracting one bond of each of the copy tensors together.

Many tensor network topologies are designed to represent the low-energy states of physically realistic systems by capturing certain entanglement entropy and correlation scalings of the state generated by the network [166, 167, 168, 169]. First note that the Schmidt decomposition may be regarded as a SVD of the matrix C_{ij} of the coefficients that form $|\psi\rangle$ [168]

$$\begin{aligned} |\psi\rangle &= \sum_{ij} C_{ij} |i_A\rangle |j_B\rangle = \sum_{ij\alpha_1\alpha_2} V_{i\alpha_1} S_{\alpha_1\alpha_2} U_{\alpha_2 j} |i_A\rangle |j_B\rangle \\ &= \sum_{ij\alpha} \lambda_\alpha V_{i\alpha} |i_A\rangle U_{\alpha j} |j_B\rangle = \sum_{\alpha} \lambda_\alpha |s_\alpha^A\rangle |s_\alpha^B\rangle , \end{aligned} \tag{4.1}$$

where λ_α are the singular values, which are equal to the Schmidt coefficients. By the definition of entanglement entropy[1], the entanglement entropy is [168]

$$E(A,B) = -\sum_{\alpha=1}^{m} |\lambda_\alpha|^2 \log(|\lambda_\alpha|^2) \leq \log(m), \tag{4.2}$$

where the inequality is saturated when all singular values are $1/\sqrt{m}$. Therefore, the bipartite entanglement entropy is upper bounded by the logarithm of the dimension of the virtual bond connecting the two bipartitions. Moreover, there can be in general a set of virtual bonds connecting the two bipartitions, e.g., the inner patch of a 2D lattice of the projected

[1]For a pure state defined by the joint density matrix ρ_{AB} with reduced density matrices ρ_A and ρ_B corresponding to the bipartitions A and B, the entanglement entropy is defined as the von-Neumann entropy of ρ_A, or equivalently of ρ_B as $E(A,B) = -\operatorname{Tr}[\rho_A \log(\rho_A)]$

entangled pair states (PEPS) and the outer patch are interfaced by a rectangle, along which all virtual bonds are responsible for the bipartite entanglement entropy. It is then readily seen that MPS, tree tensor network (TTN) (see Sec. 4.2 for more details)[2], and PEPS all have some *boundary* as the bipartition interface, as opposed to the *volume*, i.e., two points (boundary of a line segment), are the interface for bipartitions of an MPS, and a rectangle (boundary of a finite plane) is the interface for bipartitions of a PEPS. Therefore, they capture boundary-law scaling of bipartite entanglement entropy. This is in sharp contrast to a Haar-random quantum state whose bipartite entanglement entropy has a volume-law scaling [170].

The density matrix renormalization group (DMRG) technique can employ MPS ansatz to approximately generate the ground states of common Hamiltonians because these ground states are not Haar-randomly sampled, but instead tend to have boundary-law bipartite entanglement. Such scaling patterns have been proven for the ground states of 1D gapped quantum systems [171], and for harmonic lattice systems of arbitrary dimension [172]. They have also been conjectured to exist in the ground states of most local, gapped quantum systems regardless of dimension [173].

To bipartition a multi-scale entanglement renormalization ansatz (MERA) (see Sec. 4.2 for more details) is different—we need to cut a number of virtual bonds that is proportional to $\log(L)$ (L is the size of the system), creating a logarithmic correction to the boundary-law scaling, which is in turn found in many critical-phase Hamiltonians [174].

4.2 Background on Tensor Network Quantum Machine Learning

Some tensor networks allow for interpretations of coarse-grained states at increasing levels of the network as a renormalization group or scale transformation that retains information necessary to understand the physics on longer length scales [175, 176]. This motivates the usage of such networks to perform discriminative tasks, in a manner similar to classical machine learning (ML) using neural networks with layers like convolution and pooling that perform sequential feature abstraction to reduce the dimension and to obtain a hierarchical representation of the data [177, 178]. In addition to applying TNs such as the tree tensor network (TTN) [179] and the multiscale entanglement renormalization ansatz (MERA) [180] for quantum-inspired tensor network ML algorithms [181, 182, 183], there have been efforts to variationally train the generic unitary nodes in TNs to perform quantum machine learning (QML) on data-encoded qubits. The unitary TTN [184, 185] and MERA [184, 186] have been explored for this purpose mindful of feasible implementations, such as normalized input states, on a quantum computer.

Tensor network QML models are linear classifiers on a feature space whose dimension grows exponentially in the number of data qubits and where the feature map is non-linear.

[2]An MPS can be viewed as the most unbalanced TTN with all nodes on one branch of a tree.

Such models employ fully parametrized unitary tensor nodes that form a rich subset of larger unitaries with respect to all input and output qubits upon tensor contractions. They provide circuit variational ansatze more general than those with common parametrized gate sets [187, 188, 189], although their compilations into hardware-dependent native gates are more costly because of the need to compile generic unitaries.

Unitary Tree Tensor Network (TTN)

Unitary tree tensor network (TTN) is a classically tractable realization of tensor network QML models, with a topology that can be interpreted as a local coarse-graining transformation that keeps the most relevant degrees of freedom, in a sense that the information contained within each subtree is separated from those contained outside of the subtree. We focus on 1D binary trees. A generic binary TTN consists of $\log(m)$ layers of nodes where m is the number of input features, plus a layer of data qubits appended to the leaf level of the tree. A diagram of the unitary TTN is shown in Fig. 4.2 (left). Every node in a unitary TTN is forced to be a unitary matrix with respect to its input and output Hilbert spaces. Each unitary tensor entangles a pair of inputs from the previous layer. At each layer, one of the two output qubits is unobserved and also not further operated on, while the other output qubit is evolved by a node at the next layer. If the classification is binary, at the output of the last layer, namely the root node, only one qubit is measured. Accumulation of measurement statistics then reveals the confidence in predicting the binary labels associated with the measurement basis. After variationally learning the weights in the unitary nodes, we recover a quantum channel such that the information contained in the output qubits of each layer can be viewed as a coarse-grained representation of that in the input qubits, which sequentially extracts useful features of the data encoded in the data qubits. A dephased unitary TTN has local *dephasing channels* inserted between any two layers of the network, as depicted in Fig. 4.2 (right).

Multi-scale Entanglement Renormalization Ansatz (MERA)

In tensor network QML, the multi-scale entanglement renormalization ansatz (MERA) topology overcomes the drawback of local coarse-graining in unitary TTN by adding disentanglers U, which are unitaries, to connect neighboring subtrees. Its subsequent decimation of the Hilbert space by a MERA is achieved by isometries V that obey the isometric condition only in the reverse coarse-graining direction, i.e., $V^\dagger V = I'$ but $VV^\dagger \neq I$. From the perspective of discriminative QML, these unitaries correlate information from states in neighboring subtrees. We thus refer to these unitaries as entanglers.

By the design of MERA [180], the adjoint of an isometry, namely an isometry viewed in the coarse-graining direction in QML, can be naively achieved by measuring one of the two output qubits in the computational basis and post-selecting runs with measurements yielding $|0\rangle$. However, this way of decimating the Hilbert space is generally prohibitive, given the vanishing probability of sampling a bit string of all output qubits with most of them in $|0\rangle$.

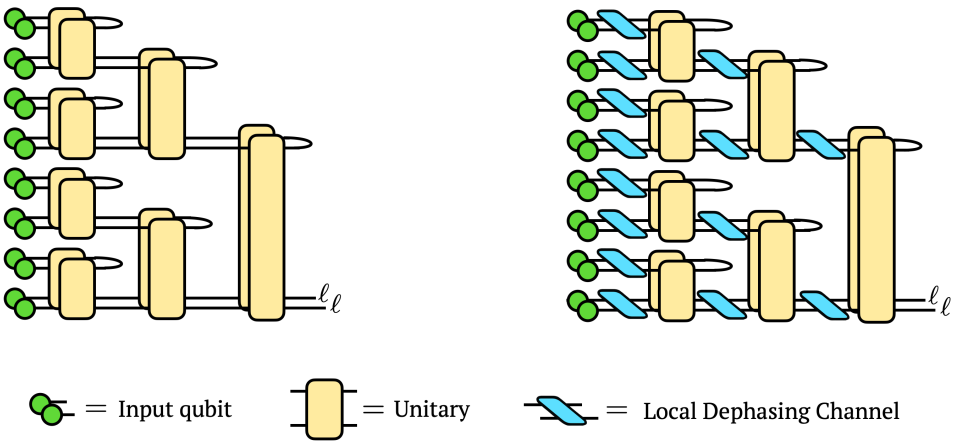


Figure 4.2: Left: A unitary TTN on eight input features encoded in the density matrices ρ_{in}'s forming the data layer, where the basis state ℓ is measured at the output of the root node. Right: Dephasing the unitary TTN is to insert dephasing channels with a dephasing rate p, assumed to be uniform across all, into the network between every layer.

Hence, operationally an isometry is replaced by a unitary node, half of whose output qubits are partially traced over, which is the same as a unitary node in the TTN. The MERA can now be understood as a unitary TTN with extra entanglers inserted before every tree layer except the root layer, such that they entangle states in neighboring subtrees, as shown in Fig. 4.3 (left). Its dephased version is similar to the dephased unitary TTN, as depicted in Fig. 4.3 (right).

Probabilistic Graphical Models

Let a set of vertices and an edge set of ordered pairs of vertices form a directed graph $G = (V, E)$, and let $X = \{X_v\}, \forall v \in V$ be a set of discrete random variables indexed by the vertices. Let $\text{pa}(v)$ or $X_{\text{pa}(v)}$ denote the set of parent vertices/variables each of which has an edge directed towards v. A directed edge represents some conditional probability of the variable on its parent. We say that X is a discrete Bayesian network (a.k.a. belief network) with respect to G if G is acyclic, namely, it is a directed acyclic graph (DAG), or equivalently if the joint probability mass function of X can be written as a product of the individual probability mass functions conditioned on their parent variables, i.e., $P(X) = \prod_{v \in V} P(X_v | X_{\text{pa}(v)})$. An undirected graphical model (UGM) is a type of probabilistic graphical model that represents the conditional dependencies between variables through an undirected graph. One of the strengths of UGMs is their ability to capture the symmetrical relationships between variables.

It was shown by Robeva et al. [190] in Theorem 2.1 that the data defining a UGM is equivalent to that defining a tensor network (TN) with non-negative nodes, but with dual graphical notations that interchange the roles of nodes and edges. Hence, we have discrete

UGM=non-negative TN, where = represents that the two classes of model can produce the same probability distribution using the same number of parameters, i.e., they are equally expressive.

The Born machine (BM) [191, 164], which models a probability mass function as the absolute value squared of a complex function, is a family of more general probabilistic models built from TNs that arise naturally from the probabilistic interpretation of quantum mechanics. The locally purified state (LPS), first discussed by Glasser et al. [191] and generalized by Miller et al. [164], adds to each node in a BM a purification edge, allowing it to represent the most general family of quantum-inspired probabilistic models. Glasser et al. [191] showed that LPS is more expressive than BM, i.e., LPS>BM.

The decohered Born Machine (DBM) was introduced by Miller et al. [164], which adds to a subset of the virtual bonds BM decoherence edges that fully dephase the underlying density matrices. A BM all of whose virtual bonds are decohered is called a fully-DBM. Miller et al. [164] showed that fully decohering a BM gives rise to a discrete UGM, and conversely, any subgraph of a discrete UGM can be viewed as the fully-decohered version of some BM. Hence, we have fully-DBM=discrete UGM.

Theorem 3 and 4 by Miller et al. [164] showed that any DBM can be viewed as an LPS, and vice versa [164], i.e., LPS=DBM, since each purification edge joining a pair of LPS cores can be expressed as a larger network of copy tensors, and each decoherence edge of a DBM can be absorbed into nearby pair of tensors and form a purification edge. Following this view of LPS=DBM and the fact that LPS>BM, one arrives at DBM>BM, which can also be understood as BM being a special case of DBM with an empty set of decohered edges added.

A summary of the relative expressiveness is given in Tab. 4.1.

Table 4.1: The relative expressiveness, defined as the probability distributions a model can produce with the same number of parameters, among the discrete graphical model (UGM), the tensor network (TN) with non-negative nodes, the Born machine (BM), the decohered Born machine (DBM), and the locally purified state (LPS).

Relative Expressiveness	Ref.
discrete UGM = non-negative TN	[190]
fully-DBM = discrete UGM	[164]
LPS > BM	[191]
LPS = DBM > BM	[164]

The unitary TTN and the MERA, dephased or not, are DBMs or equivalently LPSs. Each partial tracing in them is represented by a purification edge, while each dephasing channel acting on the input of a unitary node in them can be viewed as a larger unitary node contracting with some environment node and the input node, before tracing out the environment degree of freedoms using a purification edge. Each of the tensor networks produces a normalized joint probability once the data nodes are specified with normalized

quantum states and the readout node is specified with a basis state. Fully dephasing every virtual bond in the network gives rise to a fully-DBM, which can be also viewed as a discrete UGM in the dual graphical picture. We describe in Sec. 4.4 that, by directly taking into account the effect of the partial tracing or the purification, the fully dephased networks can also be viewed as Bayesian networks via some directed acyclic graphs (DAGs).

4.3 Decohering Tensor Network Quantum Machine Learning Models

In this study, we focus on discriminative QML. We investigate and numerically quantify the competing effect between decoherence and increasing bond dimension of two common tensor network QML models, namely the unitary TTN and the MERA. By removing the off-diagonal elements, i.e., the coherence, from the density matrix of a quantum state, we reduce its representation down to a classical probability distribution over a given basis. The evolution through the unitary matrices at every layer of the model, together with the full dephasing of the density matrix at input and output, then becomes successive Bayesian updates of classical probability distributions, thus removing the quantumness of the model. This process can occur between any two layers of the unitary TTN or the MERA, and should in principle reduce the amount of information or representative flexibility available to the classification algorithm. However, as we add and increase the number of ancillas and accordingly increase the virtual bond dimension of the tensor networks, this diminished expressiveness may be compensated by the increased dimension of the classical probability distributions and their conditionals, manifested in the increasing number of diagonals intermediate within the network, as well as by the increased sized of the stochastic matrices encapsulated by the corresponding Bayesian networks in the fully dephased limit. The possibility that an increased bond dimension fully compensates for the decoherence of the network would indicate that the role of coherence in QML is not essential and it offers no unique advantage, whereas a partial compensation provides insights into the trade-off between adding ancillas and increasing the level of decoherence in affecting the network performance, and therefore offers guidance in determining the number of noisy ancillas to be included in NISQ-era [104] implementations.

Dephasing Qubits after Unitary Evolution

A dephasing channel with a rate $p \in (0, 1]$ on a qubit is obtained by tracing out the environment after the environment scatters off of the qubit with some probability p. We denote the dephasing channel on a qubit with a dephasing rate p as $\mathcal{E}$, such that

$$\mathcal{E}[\rho] = (1 - \frac{1}{2}p)\rho + \frac{1}{2}p\sigma_3\rho\sigma_3 = \sum_{ij}(1-p)^{1-\delta_{ij}} \langle i|\rho|j\rangle |i\rangle\langle j| = \sum_{ij}(1-p)^{1-\delta_{ij}}\rho_{ij}|i\rangle\langle j|, \tag{4.3}$$

where the summation goes from 0 to 1 for every index hereafter unless specified otherwise, whose effect is to damp the off-diagonal entries of the density matrix by $(1-p)$. The operator-sum representation of $\mathcal{E}[\rho]$ can be written as with the two Kraus operators[3],

$$K_0 = \sqrt{1-\frac{p}{2}}I, \quad K_1 = \sqrt{\frac{p}{2}}\sigma_3, \tag{4.4}$$

defined such that $\mathcal{E}[\rho] = \sum_i K_i \rho K_i^\dagger$ and $\sum_i K_i^\dagger K_i = I$. Assuming local dephasing on each qubit, the dephasing channel on the density matrix ρ of m qubits, entangled or not, is given by

$$\mathcal{E}[\rho] = \sum_{i_1,\ldots,i_m} \left(\bigotimes_{n=1}^{m} K_{i_n}\right) \rho \left(\bigotimes_{n=1}^{m} K_{i_n}^\dagger\right). \tag{4.5}$$

If we allow a generic unitary U to act on $\mathcal{E}[\rho]$ for a single qubit, we have the purity of the resultant state given by

$$\mathrm{Tr}\left[\left(U\mathcal{E}[\rho]U^\dagger\right)^2\right] = \mathrm{Tr}\left[\left(\left(1-\frac{p}{2}\right)\rho + \frac{p}{2}\sigma_3\rho\sigma_3\right)^2\right] = \mathrm{Tr}\left(\rho^2\right) - 4p\rho_{01}^2\left(1-\frac{p}{2}\right) \leq \mathrm{Tr}\left(\rho^2\right), \tag{4.6}$$

where we used Eq. (4.3) in the first line. Therefore, in a given basis, successive applications of a dephasing channel and generic unitary evolution decrease the purity of any input quantum state, until the state becomes maximally mixed[4]. Successively applying the dephasing channel alone decreases the purity of the state until it becomes fully decohered, namely diagonal in its density operator in a given basis. It is thus a process in which quantum information of the input is irreversibly and gradually (for $p < 1$) lost to the environment until the state becomes completely describable by a discrete classical probability distribution.

Dephasing Product-state Encoded Input Qubits

When inputting data into a tensor network, it is common to featurize each sample into a product state, or a rank-one tensor. The density matrix of such a state with m features is given by $\rho = \bigotimes_{n=1}^{m} |f^{(n)}\rangle\langle f^{(n)}| = \bigotimes_{n=1}^{m} \rho^{(n)}$, where $|f^{(n)}\rangle$ is a state of dimension d that encodes the nth feature. Assuming local dephasing on each data qubit, it is expected that the product state density matrix after dephasing is the product state of the dephased component density matrix, i.e., $\mathcal{E}[\rho] = (\bigotimes_{n=1}^{m} \mathcal{E}^{(n)})[\bigotimes_{n=1}^{m} \rho^{(n)}] = \bigotimes_{n=1}^{N} \mathcal{E}^{(n)}[\rho^{(n)}]$.

In the context of our tensor network classifier, the effect of dephasing can be seen by considering just a single feature. If we normalize this feature such that its value is $x^{(n)} \in [0,1]$, then we can utilize the commonly-used qubit encoding [192, 193, 95] to encode this classical feature into a qubit as

$$|f^{(n)}\rangle = \begin{bmatrix} \sin\left(\frac{\pi}{2}x^{(n)}\right) \\ \cos\left(\frac{\pi}{2}x^{(n)}\right) \end{bmatrix}, \tag{4.7}$$

[3] A more commonly-used, but less computationally efficient in terms of Eq. (4.5), representation uses three Kraus operators: $K_0 = \sqrt{1-p}I$ and $K_{1/2} = \frac{\sqrt{p}}{2}(I \pm \sigma_3)$ such that $\mathcal{E}[\rho] = \sum_{i=0}^{2} K_i \rho K_i^\dagger$ and $\sum_{i=0}^{2} K_i^\dagger K_i = I$.

[4] Unitary evolution on the d-dimensional maximally mixed states, which are the only rotationally invariant states, does not produce coherence.

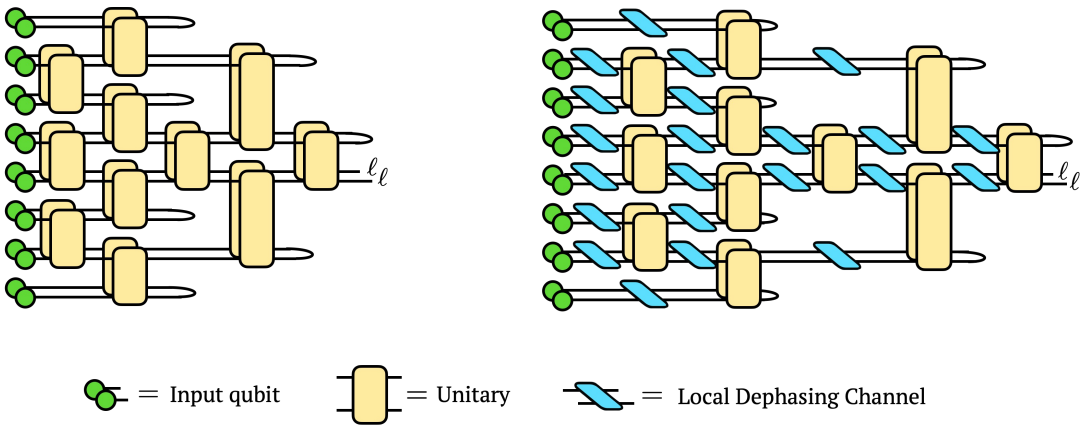


Figure 4.3: Left: A MERA on eight input features encoded in the ρ_{in}'s forming the data layer, where the basis state ℓ is measured at the output of the root node. Right: Dephasing the MERA is to insert dephasing channels with a dephasing rate p, assumed to be uniform across all, into the network between every layer.

respectively. A notable property of these encodings is that the elements of $|f^{(n)}\rangle$ are always positive, so there is a one-to-one mapping between $|\langle i^{(n)}|f^{(n)}\rangle|^2$ and $\langle i^{(n)}|f^{(n)}\rangle$ for all $i^{(n)}$. This means that every element of $\rho^{(n)} = |f^{(n)}\rangle\langle f^{(n)}| \equiv \rho$ can be written as a function of probabilities $\lambda_0^{(n)} \equiv \lambda_0$ and $\lambda_1^{(n)} \equiv \lambda_1$, where

$$\rho_{00} = \lambda_0, \quad \rho_{01} = \rho_{10} = \sqrt{\lambda_0\lambda_1}, \quad \rho_{11} = \lambda_1. \tag{4.8}$$

Using Eq. (4.16), we get

$$\lambda_0' = |U_{00}|^2\lambda_0 + |U_{01}|^2\lambda_1 + 2\sqrt{\lambda_0\lambda_1}\Re(U_{00}U_{01}) \tag{4.9}$$

$$\lambda_1' = |U_{11}|^2\lambda_1 + |U_{10}|^2\lambda_0 + 2\sqrt{\lambda_0\lambda_1}\Re(U_{10}U_{11}), \tag{4.10}$$

where it is clear that the new probabilities λ_i' are non-linear functions of the old probabilities λ_j. Specifically, there is a dependence on $\sqrt{\lambda_0\lambda_1}$. Such non-linear functions cannot be generated by a stochastic matrix acting on $\text{diag}(\rho^{(n)})$, since the off-diagonal $\sqrt{\lambda_0\lambda_1}$ terms will be set to zero. By fully dephasing the input state before acting the unitary, the fully dephased output is less expressive in the sense that we lose the regressor $\sqrt{\lambda_0\lambda_1}$. But knowing the relative phase of the encoding, this lost regressor does not contain any extra information than the regressors λ_0 and λ_1, so in that sense the information content of the encoding is unaffected by the dephasing.

Impact on Regressors by Dephasing

To understand the dephasing effect on the linear regression induced by the unitary TTN network topology, it is illuminating to study the evolution of $\text{Tr}_A(U\mathcal{E}[\rho]U^\dagger)$ which is un-

dertaken by a unitary node acting on a pair of dephased input qubits followed by a partial tracing over one of the output qubits. The diagonals of the output density matrix before partial tracing, i.e., the diagonals of $U\mathcal{E}[\rho]U^\dagger$, are

$$\begin{aligned}\rho'_{ii} = |U_{i0}|^2\rho_{00} + |U_{i1}|^2\rho_{11} + |U_{i2}|^2\rho_{22} + |U_{i3}|^2\rho_{33} + \\ 2(1-p)\left[\Re(U_{i1}U^*_{i0}\rho_{10}) + \Re(U_{i2}U^*_{i0}\rho_{20}) + \right. \\ \left. \Re(U_{i3}U^*_{i1}\rho_{31}) + \Re(U_{i3}U^*_{i2}\rho_{32})\right] + \\ 2(1-p)^2\left[\Re(U_{i3}U^*_{i0}\rho_{30}) + \Re(U_{i2}U^*_{i1}\rho_{21})\right],\end{aligned} \tag{4.11}$$

for $i \in \{0,1,2,3\}$, where every diagonal term is a linear regression on all elements of input ρ with regression coefficients set by the unitary matrix elements $U_{ik}, k \in \{0,1,2,3\}$. We note that terms such as the $\Re(U_{i1}U^*_{i0}\rho_{10}) = U_{i0}U^*_{i1}\rho_{01} + U_{i1}U^*_{i0}\rho_{10}$ are each composed of two regressors. In particular, the dephasing suppresses some of the regressors by a factor of $(1-p)$ or $(1-p)^2$. Since the norm of each element in U and $U^\dagger$ is upper bounded by one, the norm of the regression coefficients is suppressed by these factors induced by dephasing. The suppression is stronger by a factor of $(1-p)^2$ for regressors that are anti-diagonals of the input density matrix, i.e., ρ_{30} and ρ_{21}. While the regression described above is to obtain the diagonals of the output density matrix, the regression to obtain off-diagonals of the output density matrix has a similar pattern of suppression of certain regressors.

This suppression of regression coefficients is carried over to the reduced density matrix, which can be written as

$$\mathrm{Tr}_2(\rho') = \begin{bmatrix}\rho'_{00}+\rho'_{11} & \rho'_{02}+\rho'_{13} \\ \rho'_{20}+\rho'_{31} & \rho'_{22}+\rho'_{33}\end{bmatrix}. \tag{4.12}$$

When the input pair of qubits ρ is a product state of two data qubits, we have

$$\rho = \rho^{(1)} \otimes \rho^{(2)} \equiv \begin{bmatrix}\lambda_0 & \sqrt{\lambda_0\lambda_1} \\ \sqrt{\lambda_0\lambda_1} & \lambda_1\end{bmatrix} \otimes \begin{bmatrix}\mu_0 & \sqrt{\mu_0\mu_1} \\ \sqrt{\mu_0\mu_1} & \mu_1\end{bmatrix}, \tag{4.13}$$

where the λ's and μ's are defined like Eq. (4.8) for the two data qubits $\rho^{(1)}$ and $\rho^{(2)}$. Substituting Eq. (4.13) into Eq. (4.11) and (4.12), we see that all regressors containing $\sqrt{\mu_0\mu_1}$ or $\sqrt{\lambda_0\lambda_1}$ are suppressed by a factor of $(1-p)$ after the first-layer unitary, while the regressor $\sqrt{\lambda_0\lambda_1\mu_0\mu_1}$ is suppressed by a factor of $(1-p)^2$. The output density matrix elements then become the regressors for regressions performed by subsequent upper layers, as follows.

For unitary TTN without *ancillas*, Eq. (4.11) and (4.12) are carried over to the output of every layer of the network, since there is no entanglement in the input pair of qubits. However, at the upper layers, the regression onto the output density matrix element has regressors already composed of terms that were suppressed in previous layers, as described above for $\rho \to \rho'$. Viewing the regressors at the input of the last layer, the suppression on most of them by some power of $(1-p)$ resembles the concept of regularization in regressions but does not involve a penalty term on the coefficient norm in the loss function.

In cases where there can be entanglement in each of the input qubits, such as the intermediate layers in a MERA or in a unitary TTN with ancillas, the pattern of suppressing

certain regressors is similar, where the coherence of the input is suppressed by some power of $(1-p)$. In particular, the regressors on the anti-diagonals are most strongly suppressed by a factor of $(1-p)^m$ where m is the number of input qubits.

4.4 fully dephased Unitary Tensor Networks

When the network is fully dephased at every layer, all of the off-diagonal regressors are removed. Each diagonal term of the output density matrix then becomes a regression on only the diagonals of the input density matrix. In Sec. 4.4, we show that in this situation each node of the unitary tensor network U_{ij} reduces to a unitary-stochastic matrix $M_{ij} \equiv |U_{ij}|^2$. When the output of the unitary node is partially traced over, the overall operation is equivalent to a singly stochastic matrix $S_{i_B j} \equiv \sum_{i_A} |U_{i_A i_B j}|^2$, where i_A enumerates the traced-over part of the system. The tensor network QML model then reduces to a classical Bayesian network (see Sec. 4.2) with the joint probability factorization Eq. (4.21) which shall be presented in Sec. 4.4 and 4.4.

Fully-dephasing Qubits after Unitary Evolution

To fully dephase a quantum state, we simply choose a basis to represent the density matrix and then set all off-diagonal elements of the matrix to zero, leaving the diagonal elements unchanged. If we represent the fully-dephasing ($p = 1$) superoperator as $\mathcal{D}$, then

$$\mathcal{D}[\rho] = \sum_i \langle i|\rho|i\rangle \, |i\rangle\langle i| = \sum_i \rho_{ii} |i\rangle\langle i|. \tag{4.14}$$

For convenience, we adopt the notation $\lambda_i \equiv \rho_{ii}$, where the λ_i can be identified as probabilities from some discrete distribution. If we allow a generic unitary U to act on ρ before it is fully dephased, then we have

$$\mathcal{D}[U\rho U^\dagger] = \sum_i \langle i|U\rho U^\dagger|i\rangle \, |i\rangle\langle i| = \sum_{ijk} \rho_{jk} \langle i|U|j\rangle \langle k|U^\dagger|i\rangle \, |i\rangle\langle i|, \tag{4.15}$$

so that the new probabilities λ'_i encoded in the fully dephased state are given by

$$\lambda'_i = \mathcal{D}[U\rho U^\dagger]_{ii} = \sum_{jk} \rho_{jk} \langle i|U|j\rangle \langle k|U^\dagger|i\rangle = \sum_{jk} \rho_{jk} U_{ij} U^*_{ik} \tag{4.16}$$

From Eq. (4.16), we can see that each probability is a function of the entire density matrix, along with the elements of U. If ρ is assumed to be fully dephased already, then $\rho_{jk} = \lambda_j \delta_{jk}$ and therefore

$$\lambda'_i = \sum_{jk} \lambda_j \delta_{jk} U_{ij} U^*_{ik} = \sum_j \lambda_j |U_{ij}|^2 = \sum_j M_{ij} \lambda_j. \tag{4.17}$$

By the unitarity of U, $M_{ij} \equiv |U_{ij}|^2$ is doubly stochastic, i.e., $\sum_i M_{ij} = \sum_i |U_{ij}|^2 = \mathbb{1}_j$ and $\sum_j M_{ij} = \sum_j |U_{ij}|^2 = \mathbb{1}_i$, which maps the old probabilities λ to new probabilities λ' that are

normalized, i.e., $\sum_i \lambda'_i = \sum_{ij} M_{ij}\lambda_j = \sum_j \mathbb{1}_j \lambda_j = 1$. Such doubly stochastic matrices M that correspond to some unitaries are called unitary-stochastic matrices. For $N \leq 2$, all $N \times N$ doubly stochastic matrices are also unitary-stochastic. But unitary-stochastic matrices form a proper subset of doubly stochastic matrices for $N \geq 3$[5] [195, 194].

Fully-dephasing a Reduced Density Matrix after Unitary Evolution

In some tensor networks such as the TTN, the effective size of the feature space is reduced by tracing over some of the degrees of freedom after each layer. The combined effects of the unitary layer and partial trace produce a quantum channel, whose output is then fully dephased. If we partition the Hilbert space of an input density matrix ρ into parts A and B, then the outputs λ'_{i_B} after tracing over part A are given by

$$\begin{aligned}
\lambda'_{i_B} &= \left[\mathrm{Tr}_A\left(\mathcal{D}[U\rho U^\dagger]\right)\right]_{i_B i_B} \\
&= \left[\sum_{i_A i_B jk} \mathrm{Tr}_A\left(\rho_{jk}\langle i_A i_B|U|j\rangle\langle k|U^\dagger|i_A i_B\rangle |i_A\rangle\langle i_A||i_B\rangle\langle i_B|\right)\right]_{i_B i_B} \\
&= \sum_{i_A jk} \rho_{jk}\langle i_A i_B|U|j\rangle\langle k|U^\dagger|i_A i_B\rangle \,\mathrm{Tr}\left(|i_A\rangle\langle i_A|\right) \\
&= \sum_{jk}\rho_{jk}\sum_{i_A} U_{i_A i_B j}U^*_{i_A i_B k}.
\end{aligned} \tag{4.18}$$

We can again see that the output diagonals depend on all elements of ρ and U. If ρ is already fully dephased, then we have

$$\lambda'_{i_B} = \sum_{jk}\lambda_j\delta_{jk}\sum_{i_A} U_{i_A i_B j}U^*_{i_A i_B k} = \sum_j \lambda_j \sum_{i_A}|U_{i_A i_B j}|^2 = \sum_j S_{i_B j}\lambda_j, \tag{4.19}$$

where $S_{i_B j} \equiv \sum_{i_A}|U_{i_A i_B j}|^2$ is a rectangular singly stochastic matrix with respect to index i_B only, i.e., $\sum_{i_B} S_{i_B j} = \sum_{i_A i_B}|U_{i_A i_B j}|^2 = \mathbb{1}_j$. It again maps the old probabilities λ to new probabilities λ' which are normalized, i.e., $\sum_{i_B}\lambda'_{i_B} = \sum_{i_B j} S_{i_B j}\lambda_j = \sum_j \mathbb{1}_j\lambda_j = 1$. We remark that the output index i_B runs from 1 to $\dim(B)$, while the input index j runs from 1 to $\dim(A)\cdot\dim(B)$, and the Bayesian update by this singly stochastic matrix applies only in the coarse-graining direction.

Fully-dephasing the Unitary TTN

Dephasing a unitary TTN is to apply local dephasing channels on each pair of output bonds before contracting with the node at the next layer, as shown in Fig. 4.2 (right). In terms of the underlying density matrix, the dephasing channel is to apply Eq. (4.5) to the bonds,

[5]The dimension of the parameter space for $N \times N$ unitary-stochastic matrices is $(N-1)^2$ as for doubly stochastic matrices. The parameter space covered by unitary-stochastic matrices is, however, in general, smaller than that covered by doubly stochastic matrices [194].

each of which may represent a higher-dimensional state if there are ancilla qubits added as discussed in Sec. 4.5. We note that assuming local dephasing, there is no need to dephase before partially tracing out some generally entangled qubits out of the unitary TTN node, say tracing over part A of the output system AB, since there exists a U_{AE} on $\rho_{AB} \otimes \rho_E$ by the definition of dephasing such that

$$\mathrm{Tr}_A\left(\mathcal{E}_A\left[\rho_{AB}\right]\right) = \mathrm{Tr}_A\left[\mathrm{Tr}_E\left(U_{AE}\rho_{AB}\otimes\rho_E U_{AE}^\dagger\right)\right] = \mathrm{Tr}_A(\rho_{AB}). \tag{4.20}$$

A diagram of the dephased unitary TTN is shown in Fig. 4.2 (right).

As shown in Sec. 4.4, fully decohering after partially tracing out every composite node of a unitary TTN leads to a TTN composed of nodes each of which is a rectangular singly stochastic matrix S (reduced from a unitary-stochastic matrix), acting on a vector of the diagonals of a density matrix, that only preserves the normalization in the coarse-graining direction. The fully dephased TTN then exhibits a chain of conditional probabilities and can be interpreted as successive Bayesian updates across layers. A diagram using the third-order copy tensors (see Sec. 4.1) to fully dephase the unitary TTN is shown in Fig. 4.4 (left), and the dual graphical picture as a Bayesian network is depicted in Fig. 4.4 (right).

Formally, a fully dephased unitary TTN can be viewed as a discrete Bayesian network via a DAG with input quantum states as parent variables. In other words, the Bayesian network provides a dual graphical formulation of the fully dephased unitary TTN, with the graph edges functioning as the tensor nodes while the graph vertices acting as the virtual bonds [190, 164]. The graph vertices in the Bayesian network, which is dual to the virtual bonds in the TTN composed of stochastic matrices, represent vector variables $\lambda^{(k,j)} \equiv \mathrm{diag}(\rho^{(k,j)})$, where k and j denotes the j-indexed vertices at the kth layer of the network with 0 indexing the layer with parent variables, and ρ is the corresponding density matrix in the dual tensor network picture. We use the shorthand $\lambda^{(k)} \equiv \{\lambda^{(k,0)}, \ldots, \lambda^{(k,n_k)}\}$ to group all n_k vertices at the kth layer into a set. The output vertex of the Bayesian network stands for a readout variable ℓ specifying the basis state of the measurement. The Bayesian network then yields a joint probability once the parent variables are specified with normalized quantum states, i.e., the joint probability represented by the network can be written in the following factorized form

$$P(\lambda^{(0)}\ldots,\lambda^{(\log(m))},\ell) = P(\ell|\lambda^{(\log(m))})\prod_{k=1}^{\log(m)} P(\lambda^{(k)}|\lambda^{(k-1)})P(\lambda^{(0)}), \tag{4.21}$$

where $m \equiv n_0$ is the number of vertices at the data layer. $P(\lambda^{(k)}|\lambda^{(k-1)})$ is the conditional probability represented by the edges between the $(k-1)$th and kth layer of the Bayesian network, or equivalently by the rectangular singly stochastic matrices at the kth layer of the dual tensor network. $P(\ell|\lambda^{(\log(m))})$ is the conditional probability of obtaining the basis vector ℓ.

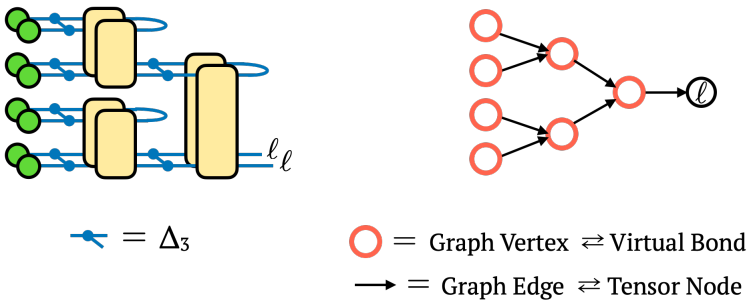


Figure 4.4: Left: Fully-dephasing a unitary TTN, where the third-order copy tensor Δ_3 is defined as $\Delta_3 = \sum_i e_i^{\otimes 3}$ with e_i the qubit basis state (see Sec. 4.1). Right: The dual graphical picture of the fully dephased unitary TTN as a Bayesian network via a directed acyclic graph (DAG). The transition matrices conditioning on each pair of input vectors are rectangular singly stochastic matrices S's reduced from some unitary-stochastic matrices.

When, for instance, the unitary TTN is fully dephased to become a Bayesian network, both schemes of adding ancillas, as described in Sec. 4.5, give rise to networks that share the same form of factorized conditional probabilities as shown in Eq. (4.21). The difference between the two schemes lies in that adding ancillas per node leads to $\lambda^{k,j}$ fixed at two dimensional $\forall k, j$, whereas adding ancillas per data qubit allows $\lambda^{k,j}$'s dimension to grow with the number of ancillas $\forall k \in \{1, \ldots, \log(m)\}, \forall j$, since increasing virtual bond dimension increases the number of diagonals.

Fully-dephasing the MERA

Similar to the fully dephased unitary TTN, the fully dephased MERA is shown in Fig. 4.5 (left), whose dual graphical formulation as a Bayesian network is shown in Fig. 4.5 (right), such that the joint probability yielded by the network upon specifying the input quantum states as the parent variables has the same factorized form as Eq. (4.21). An entangler with fully dephased input and output transforms to a unitary-stochastic matrix M, and the partially-traced-over unitary, serving as the "isometry", with fully dephased input and output transforms to a singly stochastic matrix S (reduced from a unitary-stochastic matrix) with respect to the coarse-graining direction. We note that the dimension of the vector variables dual to the output bonds of entanlgers in the tensor network picture is twice as large as other variables, since they represent correlated variables outputted by the unitary-stochastic matrices. Each of the two outgoing directed edges from these variables can be interpreted as a conditional probability conditioning on half of the support of these discrete variables.

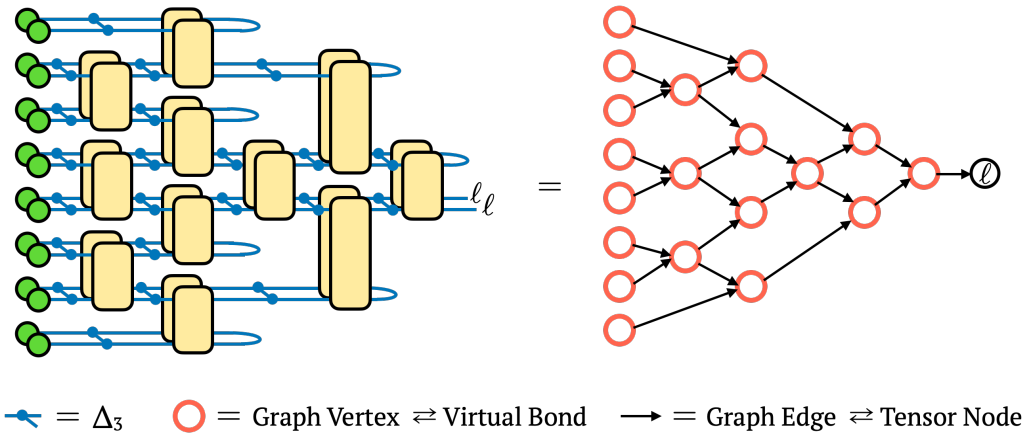


Figure 4.5: Left: Fully-dephasing a MERA. Right: Equivalently, the dual graphical picture of the fully dephased unitary TTN as a Bayesian network via a DAG, since the fully dephased MERA is a tensor network composed of unitary-stochastic matrices M's and rectangular singly stochastic matrices S's with respect to the coarse-graining direction, with input being the diagonals of the encoded qubits.

4.5 Adding Ancillas and Increasing the Virtual Bond Dimension

The Stinespring's dilation theorem [10, 196] states that any quantum channel or completely positive and trace-preserving (CPTP) map $\Lambda : \mathcal{B}(\mathcal{H}_A) \to \mathcal{B}(\mathcal{H}_B)$[6] over finite-dimensional Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$ is equivalent to a unitary operation on a higher dimensional Hilbert space $\mathcal{H}_B \otimes \mathcal{H}_E$, where $\mathcal{H}_E$ is also finite-dimensional, followed by a partial tracing over $\mathcal{H}_E$. A motivating example demonstrating directly that ancillas are necessary to allow the evolution of fully dephased input induced by a generic unitary to be as expressive as that induced by a singly stochastic matrix is presented in Sec. 4.5. In particular, the dimension of the ancillary system $\mathcal{H}_E$ can be chosen such that $\dim(\mathcal{H}_E) \leq \dim(\mathcal{H}_A)\dim(\mathcal{H}_B)$ for any Λ[7] [10]. In terms of qubits, the theorem implies that there need to be at least $2n_o$ ancilla qubits to achieve an arbitrary quantum channel between n_i input qubits and n_o output qubits. This is because the total combined number of n_i input qubits and n_a ancilla qubits should equal the total combined number of n_o output qubits and the qubits that are traced out as environment degrees of freedom. Using Stinespring's dilation theorem, we can show $2^{n_i+n_a-n_o} \leq 2^{n_i}2^{n_o}$ which leads to $n_a \leq 2n_o$.

In the scheme of adding ancillas per node in a unitary TTN, every node requires then in

[6] We denote the convex set of positive-semidefinite linear operators with unit trace, namely the set of density operators, on a complex Hilbert space $\mathcal{H}$ (thus Hermitian and bounded) as $\mathcal{B}(\mathcal{H})$.

[7] In the Stinespring's representation of such a CPTP map Λ, there exists an isometry $V : \mathcal{B}(\mathcal{H}_A) \to \mathcal{B}(\mathcal{H}_B \otimes \mathcal{H}_E)$ such that $\Lambda(\rho) = \mathrm{Tr}_E(V\rho V^\dagger), \forall \rho \in \mathcal{B}(\mathcal{H}_A)$.

principle at least two ancilla qubits to achieve an arbitrary quantum channel, because there are two input qubits coming from the previous layer and one output qubit passing to the next layer.

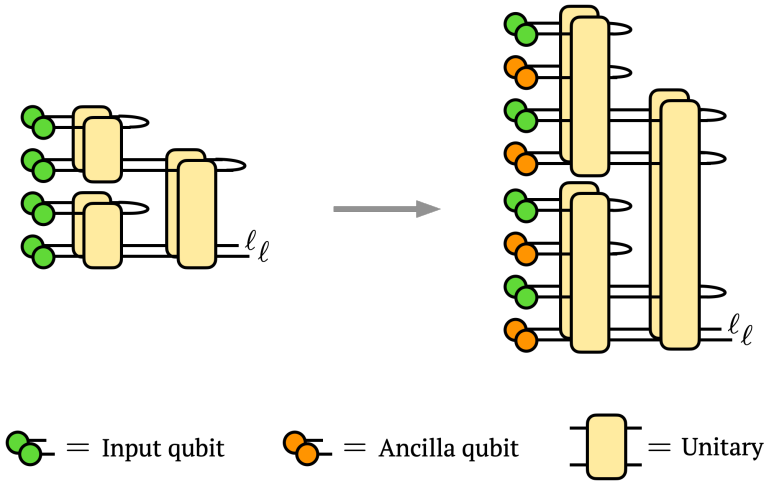


Figure 4.6: Adding one ancilla qubit, initialized to a fixed basis state, per data qubit to a unitary TTN classifying four features, with a corresponding virtual bond dimension increased to four. Only one output qubit is measured in the basis state ℓ regardless of the number of ancillas added per data qubit. We always decimate the Hilbert space by half between consecutive layers of unitary nodes.

However, in practice, we have found it more expressive to instead add ancillas to the data qubits and to trace out half of all output qubits per node before contracting with the node at the next layer. We call this the ancilla-per-data-qubit scheme. This scheme is able to achieve superior classification performance in the numerical experiment tasks that we conducted compared to the ancilla-per-unitary-node scheme described above (see details towards the end of this Section), despite the fact that the two schemes share the same number of trainable parameters when adding the same number of ancillas. A diagram of this ancilla scheme is shown in Fig. 4.6. This scheme effectively increases the virtual bond dimension of the network, which means that the network can represent a larger subset of unitaries on all input qubits.

Although the ancilla-per-data-qubit scheme achieves superior classification performance, it never produces arbitrary quantum channels at each node. To see this, for any unitary node in the first layer, the number of input qubits is $n_i = 2$, that of ancillas is $n_a = n_i k = 2k$ where $k \in \mathbb{Z}$ is the number of ancillas per data qubit, and that of output qubits passing to the next layer is $n_o = 1 + k$ such that $n_a < 2n_o, \forall a \in \mathbb{Z}$. As a result, the channels achievable via the first layer of unitaries constitute only a subset of all possible channels between its input and output density matrices. For any unitary node in subsequent layers, there are no longer any ancillas, whereas there is at least one output qubit observed or operated on later.

Consequently, the channels achievable via each layer of unitaries then also constitute only a subset of all possible channels between its input and output density matrices.

Comparing the Two Ancilla Schemes in the Unitary TTN

As shown in Tab. 4.2, adding one ancilla per data qubit and accordingly doubling the virtual bond dimension yields superior performance to adding two ancillas per unitary node, in the task of classifying 1902 8×8-compressed MNIST images each showing a digit 3 or 5. Both ancilla-added unitary TTNs are trained on 5000 samples using the Adam optimizer and validated on 2000 samples. The two ancilla schemes share the same number of trainable parameters.

Table 4.2: Average testing accuracies over five trials between adding two ancillas per unitary node and adding one ancilla per data qubit, when the dephasing rate $p = 0$ or $p = 1$, in the same classification task.

	Per unitary node	Per data qubit
$p = 0$	0.938 ± 0.001	0.972 ± 0.001
$p = 1$	0.912 ± 0.002	0.940 ± 0.002

Ancillas Are Required to Achieve Evolution by Singly Stochastic Matrices

Ancillas are necessary to allow the evolution of fully dephased input induced by a generic unitary to be as expressive as that induced by general singly stochastic matrices. Consider a singly stochastic matrix

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{4.22}$$

which maps an input state in $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ to $|0\rangle$. Note that this is naturally a mapping between fully dephased input and fully dephased output. But this mapping cannot be achieved by acting a unitary on the data qubit alone. To achieve that, we need to unitarily evolve a combined system including at least one ancilla. After tracing out the ancilla, it is possible to leave the data qubit in $|0\rangle$. Namely, $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\} \to |0\rangle \otimes |0\rangle_E$ or $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\} \to |0\rangle \otimes |1\rangle_E$ is achievable by a unitary on the combined system. Note that this is also a mapping between fully dephased input and fully dephased output naturally. Therefore, considering generic unitary evolution such as contracting with a node in the unitary TTN, it is necessary to include ancillas to achieve what can be mapped by a singly stochastic matrix between the fully dephased input and fully dephased output.

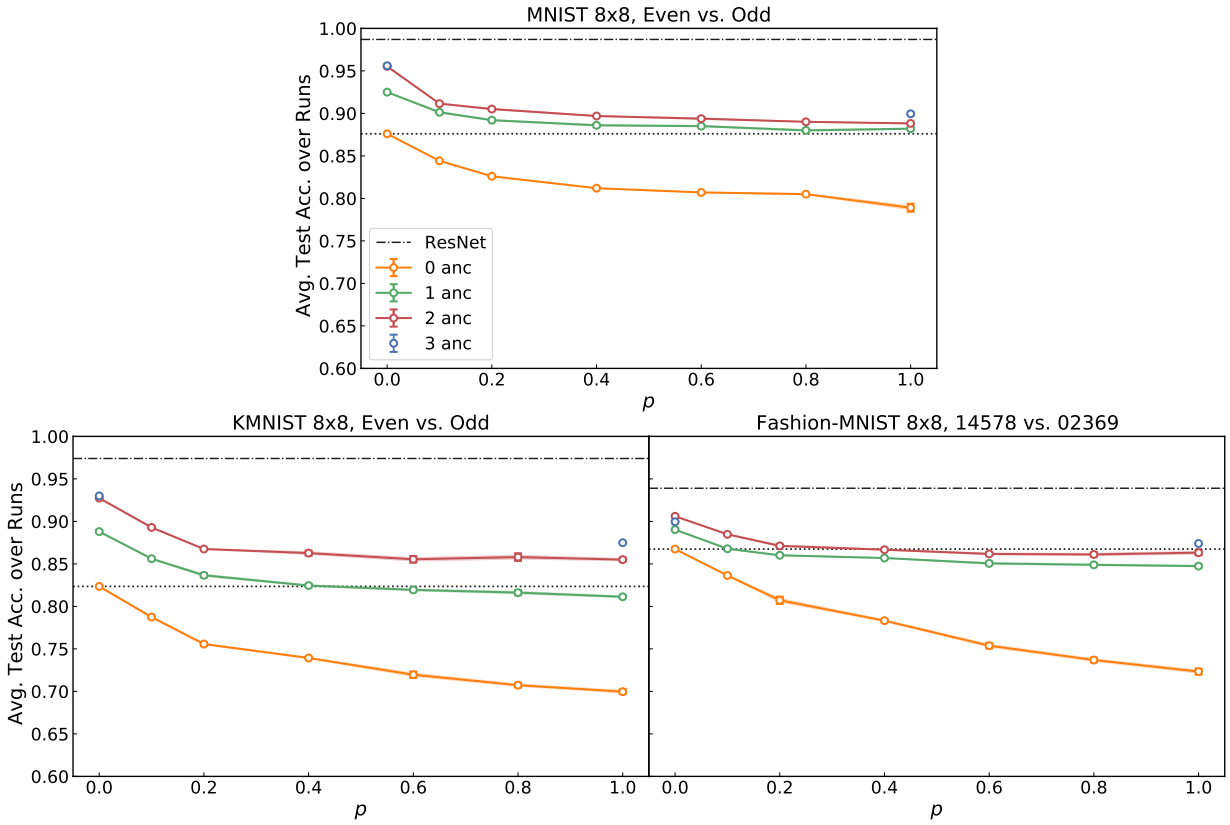


Figure 4.7: Average testing accuracy over five runs with random batching and random initialization as a function of dephasing probability p when binary-classifying 8×8 compressed MNIST, KMNIST, or Fashion-MNIST images. In each image dataset, we group the original ten classes into two, with the grouping shown in the titles. Every layer of the unitary TTN, including the data layer, is locally dephased with a probability p. Each curve represents the results from the network with a certain number of ancillas added per data qubit, with the error bars showing one standard error. The dotted reference line shows the accuracy of the non-dephased network without any ancilla.

4.6 Numerical Experiments

To demonstrate the competing effect between dephasing and adding ancillas while accordingly increasing the bond dimension of the network, we train the unitary TTN to perform binary classification on grouped classes on three datasets of different levels of difficulty[8]. Recall that n_i, n_a, and n_o respectively denote the number of input data qubits, ancillas, and output qubits, of every unitary node in the first layer of the TN. We employ TTNs with $n_i = 2$, $n_a \in \{0, n_i, 2n_i, 3n_i\}$, and $n_o = 1/2(n_i + n_a)$ for every unitary node in the first layer, and with virtual bond dimensions equal $1/2(n_i + n_a)$. We also employ MERAs with $n_i = 2$, $n_a \in \{0, n_i\}$, and $n_o = 1/2(n_i + n_a)$ for every unitary node in the first layer, and with virtual bond dimensions equal $1/2(n_i + n_a)$. The root node in either network has one output qubit measured for a binary prediction.

We vary both the dephasing probability p in dephasing every layer of the network, and the number of ancillas, which results in a varying bond dimension of the TTN. In the fully dephased limit, the unitary TTN essentially becomes a Bayesian network that computes a classical joint probability distribution (see Sec. 4.4).

In each dataset, we use a training set of 50040 samples of 8×8-compressed images and a validation dataset of 9960 samples, and we employ the qubit encoding given in Eq. (4.7). The performance is evaluated by classifying another 10000 testing samples. The unitarity of each node is enforced by parametrizing a Hermitian matrix H and letting $U = e^{iH}$. In all of our cases where the model can be efficiently simulated[9], they can be optimized with analytic gradients using the Adam optimizer [55] with respect to a categorical cross-entropy loss function, with backpropagations through the dephasing channels. Values of the hyperparameters employed in the optimizer (learning rate) and for initialization of the unitaries (standard deviations) are tabulated in Sec. 4.6. The ResNet-18 model [197], serving as a benchmark of the state-of-the-art classical image recognition model, is adapted to and trained/tested on the same compressed, grayscale images.

For the first 8×8-compressed, grayscale MNIST [198] dataset, and the second 8×8-compressed, grayscale KMNIST [199] dataset, we group all even-labeled original classes into one class and group all odd-labeled original classes into another, and perform binary classification on them. For the third 8×8-compressed, grayscale Fashion-MNIST [200] dataset, we group $0, 2, 3, 6, 9$-labeled original classes into one class and the rest into another. The binary classification performance on each of the three datasets as a function of dephasing probability p and the number of ancillas is shown for the unitary TTN in Fig. 4.7. Due to high computational costs, we simulate a three-ancilla network with p values equal to 0 and 1 only. This suffices to reveal the performance trends in both the non-decohered unitary tensor network and the corresponding Bayesian network.

There are two interesting observations to make on the results in Fig. 4.7. First, the classification performance is very sensitive to small decoherence and decreases the most

[8] Example images of the three datasets are shown in Sec. 4.6.

[9] If the model cannot be efficiently simulated, stochastic approximations such as the simultaneous perturbation stochastic approximation (SPSA) with momentum algorithm [185] can be used for training.

rapidly in the small p regime, especially in networks with at least one ancilla added. Further dephasing the network does not decrease the performance significantly, and in some cases, it does not further decrease the performance at all. A similar observation is made for the MERA (see Fig. 4.9). Second, in the strongly dephased regime where the ancillas are very noisy, adding such noisy ancillas helps the network regain performance relative to that of the non-dephased no-ancilla network. On all three datasets, the performance regained after adding two ancillas across all dephasing probabilities is comparable to the performance with the no-ancilla non-dephased network. This suggests that in implementing such unitary TTNs in the NISQ era with noisy ancillas, it is favorable to add at least two ancillas to the network and to accordingly expand the bond dimension of the unitary TTN to at least eight, regardless of the decoherence this may introduce.

However, due to the high computational costs with more than three ancillas added to the network, our experiments do not provide sufficient information about whether the corresponding Bayesian network in the fully dephased limit will ever reach the same level of classification performance as the non-dephased unitary TTN by increasing the number of ancillas. Despite this, we note that in the KMNIST and Fashion-MNIST datasets, the rate of improvement of the Bayesian network as more ancillas are added is diminishing.

Fig. 4.7 shows that when classifying the Fashion-MNIST dataset, adding three ancillas in the non-decohered network leads to a slightly worse performance than just adding two ancillas. This may be attributed to the degradation problem in optimizing complex models, which is well-known in the context of classical neural networks [197]. For neural networks, this is manifested by a performance drop in both training and testing as more layers are added, and is distinguished from overfitting where only the testing accuracy drops. In the current unitary TTN calculations, the eight-qubit unitaries that arise in the three-ancilla setting are significantly harder to optimize than the six-qubit unitaries that arise in the two-qubit setting. The optimization was unable to adequately learn the eight-qubit unitaries and thus there is a small drop in performance seen on increasing the ancilla count from two to three.

Dephasing the data layer is special compared to dephasing other internal layers within the network, since the coherence in each of the product-state data qubits has not been mixed to form the next-layer features. Since the coherences are non-linear functions of the diagonals of ρ, given the linear nature of tensor networks, it is not possible to reproduce the coherence in the data qubits in subsequent layers once the input qubits are fully dephased. To examine to what extent the observed performance decrement may be attributed to decoherence within the network as opposed to decoherence of the data qubits, we perform the same numerical experiment on the Fashion-MNIST dataset but keep the input qubits coherent without any dephasing. The result, shown in Fig. 4.8, indicates that the decoherence of the virtual bonds in the unitary TTN alone is a significant source causing the classification performance to decrease, accounting for more than half of the performance decrement.

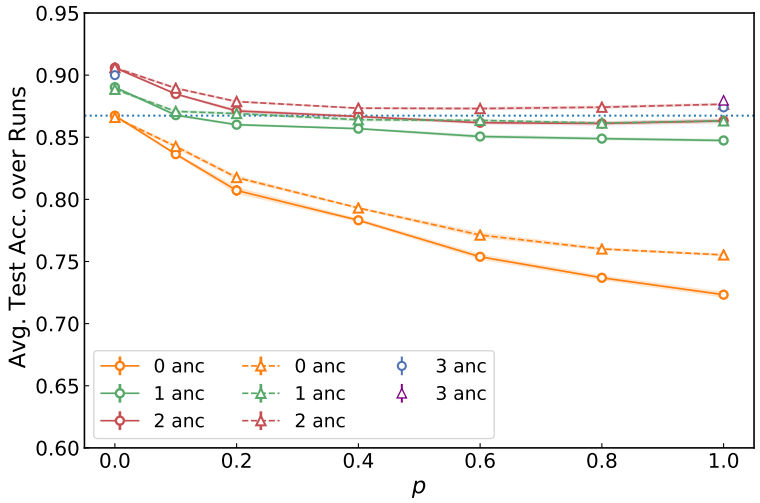


Figure 4.8: Average testing accuracy over five runs as a function of dephasing probability p when classifying 8×8 compressed Fashion-MNIST images. Each curve represents the results from the network with a certain number of ancillas added per data qubit. The circles (triangles) show the performance of the unitary TTN when every layer including (except) the data layer is locally dephased with a probability p. The dotted reference line shows the accuracy of the non-dephased network without any ancillas.

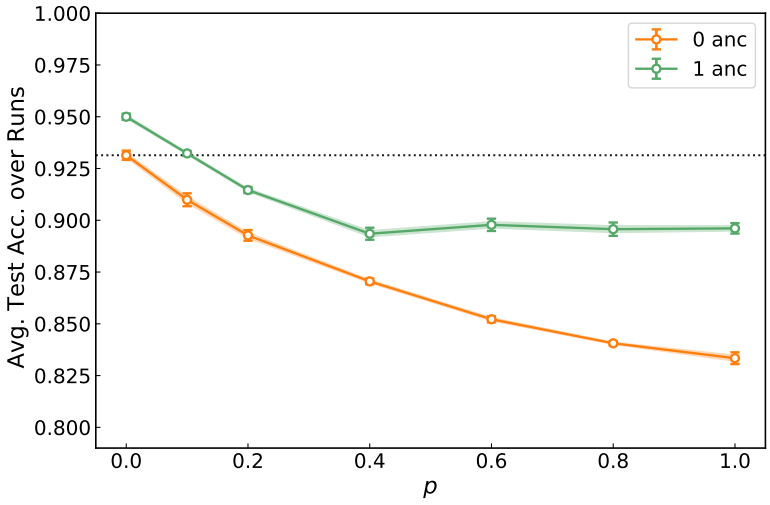


Figure 4.9: Average testing accuracy over ten runs with random batching and initialization as a function of dephasing probability p in dephasing a 1D MERA structured tensor network to classify the eight principle components of non-compressed MNIST images. Ancillas are added per data qubit. The dotted reference line shows the accuracy of the non-dephased network without any ancilla.

Datasets and Hyperparameters for the Numerical Experiments

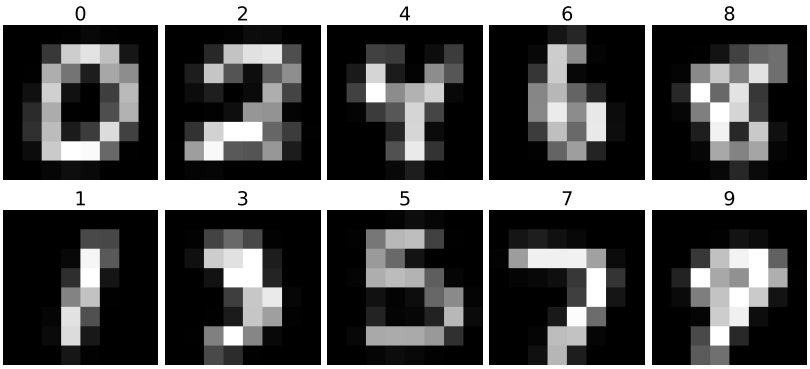
Samples from the three datasets used here are illustrated in Fig. 4.10. Compression of the images to dimension 8×8 allows tractable computation and optimization when ancillas are added to the tensor network QML models. Each pixel of an image is featurized through Eq. (4.7). The three datasets have different levels of difficulty in terms of binary classification of grouped classes, with the MNIST dataset being the easiest one while the Fashion-MNIST dataset being the most challenging.

For each dataset, the numbers of training validation, and testing samples are 50040, 9960, and 10000, respectively. The batch size used for training each model is 250. We find that initializing the Hermitian matrices around zero, or equivalently the unitaries around the identity, yields better model performance. We use random normal distributions to initialize the entries (both the real and imaginary parts) of the Hermitian matrices, with means set to 0 and standard deviation values tabulated below.

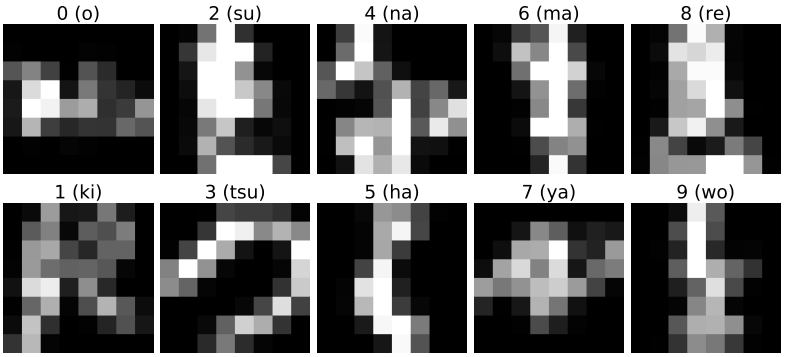
4.7 Discussion

In this study, we investigated the competition between dephasing tensor network QML models and adding ancillas to the networks, in an effort to investigate the advantage of coherence in QML and to provide guidance in determining the number of noisy ancillas to be included in NISQ-era implementations of these models. On one hand, as we increase the dephasing probability p of every layer of the network, every regressor associated with each layer of unitary nodes will have certain terms in it damped by some power of $(1-p)$. The damping cannot be offset by the regression coefficients which are given in terms of the elements of the unitary matrices. The effect of this damping of the regressors under dephasing decreases the classification accuracy of the QML model. When the network is fully dephased, these regressors are eliminated, and the tensor network QML model becomes a classical Bayesian network that is completely describable by classical probabilities and stochastic matrices. On the other hand, as we increase the number of input ancillas and accordingly increase the virtual bond dimensions of the tensor network, we allow the network to represent a larger subset of unitaries between the input and output qubits. As a result, the performance of the network improves, as demonstrated by adding up to two ancillas and a corresponding increment of the virtual bond dimension to eight in our numerical experiments. This improvement applies to all decoherence probabilities. We also find that adding more than two ancillas gives either diminishing or no improvement (Fig. 4.7). The numerical experiments are insufficient to show whether the performance of the corresponding Bayesian network can match that of the non-decohered network as more than three ancillas are added, although we did find that in the KMNIST and Fashion-MNIST datasets the rate of improvement of the Bayesian network as more ancillas are added is diminishing. It remains an open question where coherence provides any quantum advantage in QML.

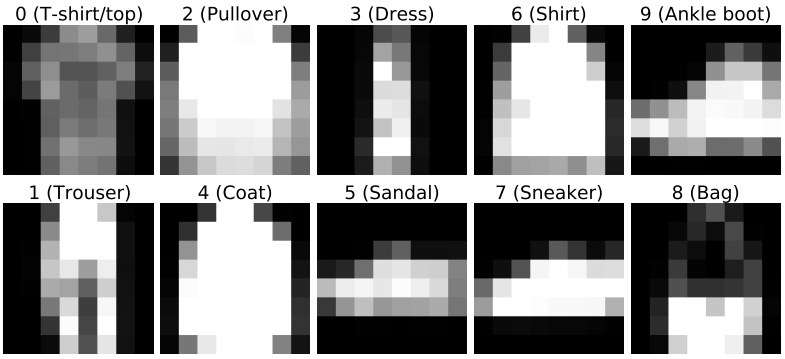
Most importantly, we find that the performance of the two-ancilla Bayesian network,



(a) 8×8-compressed MNIST images



(b) 8×8-compressed KMNIST images



(c) 8×8-compressed Fashion-MNIST images

Figure 4.10: Example images of each original class in the three datasets, with the class label shown above each example. In each dataset, the classes in the top row are grouped into one and the classes in the bottom row are grouped into another for binary classification.

namely the fully dephased network, is comparable to that of the corresponding non-decohered unitary TTN with no ancilla, suggesting that when implementing the unitary TTN, it is favorable to add at least two arbitrarily noisy ancillas and to accordingly increase the virtual bond dimension to at least eight.

We also observe that the performance of both the unitary TTN and the MERA decreases most rapidly in the small decoherence regime. With ancillas added, the performance decreases and quickly levels off at around $p = 0.2$ for the unitary TTN. The MERA with one ancilla added also exhibits this level-off performance after around $p = 0.4$. However, without any ancilla added, neither the unitary TTN nor the MERA shows a level-off performance and their performance decreases all the way until the networks are fully dephased. This contrast is an interesting phenomenon to be studied in the future.

We note that the ancilla scheme discussed in Sec. 4.5 and the theoretical analysis of the fully-decohered network presented in Sec. 4.4 are also relevant to other variational quantum ansatz states beyond tensor network QML models. For example, the analysis applies to non-linear QML models consisting of generic unitaries, such as those incorporating operations conditioned on mid-circuit measurement results of some of the qubits [186]. They may behave similarly under the competition between decoherence and adding ancillas, and it is an interesting problem for future investigation.

Bibliography

[1] Ian Convy et al. "Machine learning for continuous quantum error correction on superconducting qubits". *New Journal of Physics* 24.6 (2022), p. 063019. DOI: 10.1088/1367-2630/ac66f9.

[2] Lorenza Viola, Emanuel Knill, and Seth Lloyd. "Dynamical Decoupling of Open Quantum Systems". *Physical Review Letters* 82.12 (1999), pp. 2417–2421. DOI: 10.1103/PhysRevLett.82.2417.

[3] Adam D. Bookatz, Edward Farhi, and Leo Zhou. "Error suppression in Hamiltonian-based quantum computation using energy penalties". *Physical Review A* 92.2 (2015), p. 022317. DOI: 10.1103/PhysRevA.92.022317.

[4] Daniel A. Lidar, Isaac L. Chuang, and K. Birgitta Whaley. "Decoherence-Free Subspaces for Quantum Computation". *Physical Review Letters* 81.12 (1998), pp. 2594–2597. DOI: 10.1103/PhysRevLett.81.2594.

[5] Zhenyu Cai et al. "Quantum Error Mitigation". *arXiv:2210.00921* (2022). DOI: 10.48550/arXiv.2210.00921.

[6] Daniel A. Lidar. *Quantum Error Correction*. Cambridge University Press, 2013. DOI: 10.1017/CBO9781139034807.

[7] John Preskill. "Fault-tolerant quantum computation". *arXiv:quant-ph/9712048* (1997). DOI: 10.48550/arXiv.quant-ph/9712048.

[8] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010. DOI: 10.1017/CBO9780511976667.

[9] John Preskill. "Preskill Notes - Chapter 3: Measurements and Evolution". *Lecture Notes for Ph219/CS219* October (2018).

[10] Dennis Kretschmann, Dirk Schlingemann, and Reinhard F. Werner. "The Information-Disturbance Tradeoff and the Continuity of Stinespring's Representation". *IEEE Transactions on Information Theory* 54.4 (2008), pp. 1708–1717. DOI: 10.1109/TIT.2008.917696.

[11] Christopher J. Wood, Jacob D. Biamonte, and David G. Cory. "Tensor networks and graphical calculus for open quantum systems". *Quantum Information and Computation* 15 (2015), pp. 0579–0811. DOI: 10.48550/arXiv.1111.6950.

[12] Brecht Donvil and Paolo Muratore-Ginanneschi. "Quantum trajectory framework for general time-local master equations". *Nature Communications* 13.1 (2022). DOI: 10.48550/arXiv.1111.6950.

[13] John Preskill. "Preskill Notes - Chapter 7: Quantum Error Correction". *Lecture Notes for Ph219/CS219* October (2018).

[14] *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 452.1954 (1996), pp. 2551–2577. DOI: 10.1098/rspa.1996.0136.

[15] Daniel Gottesman. "An Introduction to Quantum Error Correction and Fault-Tolerant Quantum Computation". *arXiv:0904.2557* (2009). DOI: 10.48550/arXiv.0904.2557.

[16] A Yu Kitaev. "Quantum computations: algorithms and error correction". *Russian Mathematical Surveys* 52.6 (1997), p. 1191. DOI: 10.1070/RM1997v052n06ABEH002155.

[17] Austin G. Fowler et al. "Surface codes: Towards practical large-scale quantum computation". *Physical Review A* 86.3 (2012). DOI: 10.1103/physreva.86.032324.

[18] Kurt Jacobs and Daniel A. Steck. "A straightforward introduction to continuous quantum measurement". *Contemporary Physics* 47.5 (2006), pp. 279–303. DOI: 10.1080/00107510601101934.

[19] Alexandre Blais et al. "Cavity quantum electrodynamics for superconducting electrical circuits: An architecture for quantum computation". *Physical Review A* 69 (6 2004), p. 062320. DOI: 10.1103/PhysRevA.69.062320.

[20] Alexandre Blais et al. "Circuit quantum electrodynamics". *Reviews of Modern Physics* 93.2 (2021), p. 025005. DOI: 10.1103/RevModPhys.93.025005.

[21] Philip Krantz et al. "A quantum engineer's guide to superconducting qubits". *Applied Physics Reviews* 6.2 (2019), p. 021318. DOI: 10.1063/1.5089550.

[22] David P. DiVincenzo and Firat Solgun. "Multi-qubit parity measurement in circuit quantum electrodynamics". *New Journal of Physics* 15.7 (2013), p. 075001. DOI: 10.1088/1367-2630/15/7/075001.

[23] Charlene Ahn, Andrew C. Doherty, and Andrew J. Landahl. "Continuous quantum error correction via quantum feedback control". *Physical Review A* 65.4 (2002), p. 042301. DOI: 10.1103/PhysRevA.65.042301.

[24] Charlene Ahn, H. M. Wiseman, and G. J. Milburn. "Quantum error correction for continuously detected errors". *Physical Review A* 67.5 (2003), p. 052310. DOI: 10.1103/PhysRevA.67.052310.

[25] Charlene Ahn, Howard Wiseman, and Kurt Jacobs. "Quantum error correction for continuously detected errors with any number of error channels per qubit". *Physical Review A* 70.2 (2004), p. 024302. DOI: 10.1103/PhysRevA.70.024302.

[26] Mohan Sarovar et al. "Practical scheme for error control using feedback". *Physical Review A* 69.5 (2004), p. 052324. DOI: 10.1103/PhysRevA.69.052324.

[27] Ognyan Oreshkov and Todd A. Brun. "Continuous quantum error correction for non-Markovian decoherence". *Physical Review A* 76.2 (2007), p. 022318. DOI: 10.1103/PhysRevA.76.022318.

[28] Bradley A. Chase, Andrew J. Landahl, and Jm Geremia. "Efficient feedback controllers for continuous-time quantum error correction". *Physical Review A* 77.3 (2008), p. 032304. DOI: 10.1103/PhysRevA.77.032304.

[29] Juan Atalaya, Alexander N. Korotkov, and K. Birgitta Whaley. "Error-correcting Bacon-Shor code with continuous measurement of noncommuting operators". *Physical Review A* 102 (2 2020), p. 022415. DOI: 10.1103/PhysRevA.102.022415.

[30] Razieh Mohseninia et al. "Always-on Quantum Error Tracking with Continuous Parity Measurements". *Quantum* 4 (2020), p. 358. DOI: 10.22331/q-2020-11-04-358.

[31] Juan Atalaya et al. "Continuous quantum error correction for evolution under time-dependent Hamiltonians". *Physical Review A* 103.4 (2021), p. 042406. DOI: 10.1103/PhysRevA.103.042406.

[32] William P. Livingston et al. "Experimental demonstration of continuous quantum error correction". *arXiv:2107.11398* (2021). DOI: 10.1038/s41467-022-29906-0.

[33] Eric Dennis et al. "Topological quantum memory". *Journal of Mathematical Physics* 43 (9 2002), pp. 4452–4505. DOI: 10.1063/1.1499754.

[34] Tameem Albash and Daniel A. Lidar. "Adiabatic quantum computation". *Review of Modern Physics* 90 (1 2018), p. 015002. DOI: 10.1103/RevModPhys.90.015002.

[35] I. M. Georgescu, S. Ashhab, and Franco Nori. "Quantum simulation". *Review of Modern Physics* 86 (1 2014), pp. 153–185. DOI: 10.1103/RevModPhys.86.153.

[36] Carlos Alexandre Brasil, Felipe Fernandes Fanchini, and Reginaldo de Jesus Napolitano. "A simple derivation of the Lindblad equation". *Revista Brasileira de Ensino de Física* 35.1 (2013), pp. 01–09. DOI: 10.1590/S1806-11172013000100003.

[37] Dario Cilluffo et al. "Collisional picture of quantum optics with giant emitters". *Physical Review Research* 2 (4 2020), p. 043070. DOI: 10.1103/PhysRevResearch.2.043070.

[38] William Livingston. "Continuous Feedback on Quantum Superconducting Circuits". PhD thesis. University of California, Berkeley, 2021.

[39] Jay Gambetta et al. "Quantum trajectory approach to circuit QED: Quantum jumps and the Zeno effect". *Physics Review A* 77 (2008), p. 012112. DOI: 10.1103/PhysRevA.77.012112.

[40] Haoran Liao et al. "Machine Learning for Practical Quantum Error Mitigation". *arXiv:2309.17368* (2023). DOI: 10.48550/arXiv.2309.17368.

[41] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016.

[42] Jens Koch et al. "Charge-insensitive qubit design derived from the Cooper pair box". *Physical Review A* 76 (4 2007), p. 042319. DOI: 10.1103/PhysRevA.76.042319.

[43] Ben Criger, Alessandro Ciani, and David P. DiVincenzo. "Multi-qubit joint measurements in circuit QED: stochastic master equation analysis". *EPJ Quantum Technology* 3 (2016), p. 6. DOI: https://doi.org/10.1140/epjqt/s40507-016-0044-6.

[44] Alexander N. Korotkov. "Quantum Bayesian approach to circuit QED measurement with moderate bandwidth". *Physical Review A* 94 (4 2016), p. 042326. DOI: 10.1103/PhysRevA.94.042326.

[45] Robert E. Kass and Adrian E. Raftery. "Bayes Factors". *Journal of the American Statistical Association* 90.430 (1995), pp. 773–795. DOI: 10.2307/2291091.

[46] Havard Rue and Leonhard Held. *Gaussian Markov Random Fields: Theory and Applications*. Chapman and Hall/CRC, 2005. DOI: 10.1201/9780203492024.

[47] Colm A. Ryan et al. "Tomography via correlation of noisy measurement records". *Physical Review A* 91 (2 2015), p. 022118. DOI: 10.1103/PhysRevA.91.022118.

[48] D. S. Sivia and J. Skilling. *Data analysis: a Bayesian tutorial*. 2nd ed. Oxford University Press, 2006.

[49] W. Murray Wonham. "Some Applications of Stochastic Differential Equations to Optimal Nonlinear Filtering". *J.SIAM Series A Control* 2.3 (1964), pp. 347–369. DOI: 10.1137/0302028.

[50] Hideo Mabuchi. "Continuous quantum error correction as classical hybrid control". *New Journal of Physics* 11.10 (2009), p. 105044. DOI: 10.1088/1367-2630/11/10/105044.

[51] J. R. Norris. *Markov Chains*. 1st ed. Cambridge University Press, 1997. DOI: 10.1017/CBO9780511810633.

[52] Sergey Bravyi, Martin Suchara, and Alexander Vargo. "Efficient algorithms for maximum likelihood decoding in the surface code". *Physical Review A* 90.3 (2014), p. 032326. DOI: 10.1103/PhysRevA.90.032326.

[53] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-term Memory". *Neural computation* 9 (1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

[54] John F. Kolen and Stefan C. Kremer. "Gradient flow in recurrent nets: The difficulty of learning long-term dependencies". *A Field Guide to Dynamical Recurrent Networks*. IEEE, 2001, pp. 237–243.

[55] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". *arXiv:1412.6980* (2015). DOI: 10.48550/arXiv.1412.6980.

[56] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". *Proceedings of EMNLP*. ACL, 2014, pp. 1724–1734. DOI: 10.3115/v1/d14-1179.

[57] Richard Liaw et al. "Tune: A Research Platform for Distributed Model Selection and Training". *arXiv:1807.05118* (2018). DOI: 10.48550/arXiv.1807.05118.

[58] Sergio Blanes et al. "A pedagogical approach to the Magnus expansion". *European Journal of Physics* 31 (2010), p. 907. DOI: 10.1088/0143-0807/31/4/020.

[59] Ian Convy and K. Birgitta Whaley. "A Logarithmic Bayesian Approach to Quantum Error Detection". *Quantum* 6 (2022), p. 680. DOI: 10.22331/q-2022-04-04-680.

[60] Andre Xian Ming Chang, Berin Martini, and Eugenio Culurciello. "Recurrent Neural Networks Hardware Implementation on FPGA". *arXiv:1511.05552* (2016). DOI: 10.48550/arXiv.1511.05552.

[61] Diego Ristè et al. "Real-time processing of stabilizer measurements in a bit-flip code". *npj Quantum Information* 6.1 (2020), p. 71. DOI: 10.1038/s41534-020-00304-y.

[62] Jacob Biamonte et al. "Quantum machine learning". *Nature* 549 (7671 2017), pp. 195–202. DOI: 10.1038/nature23474.

[63] Andrew J. Daley et al. "Practical quantum advantage in quantum simulation". *Nature* 607.7920 (2022), pp. 667–676. DOI: 10.1038/s41586-022-04940-6.

[64] Frank Arute et al. "Quantum supremacy using a programmable superconducting processor". *Nature* 574 (7779 2019), pp. 505–510. DOI: 10.1038/s41586-019-1666-5.

[65] Earl T. Campbell, Barbara M. Terhal, and Christophe Vuillot. "Roads towards fault-tolerant universal quantum computation". *Nature* 549.7671 (2017), pp. 172–179. DOI: 10.1038/nature23460.

[66] Sergey Bravyi et al. "The future of quantum computing with superconducting qubits". *Journal of Applied Physics* 132.16 (2022), p. 160902. DOI: 10.1063/5.0082975.

[67] Abhinav Kandala et al. "Error mitigation extends the computational reach of a noisy quantum processor". *Nature* 567.7749 (2019), pp. 491–495. DOI: 10.1038/s41586-019-1040-7.

[68] Ewout van den Berg et al. "Probabilistic error cancellation with sparse Pauli-Lindblad models on noisy quantum processors". *Nature Physics* (2023). DOI: 10.1038/s41567-023-02042-2.

[69] Youngseok Kim et al. "Scalable error mitigation for noisy quantum circuits produces competitive expectation values". *Nature Physics* 19 (2023), pp. 752–759. DOI: 10.1038/s41567-022-01914-3.

[70] Youngseok Kim et al. "Evidence for the utility of quantum computing before fault tolerance". *Nature* 618 (2023), pp. 500–505. DOI: 10.1038/s41586-023-06096-3.

[71] Kristan Temme, Sergey Bravyi, and Jay M. Gambetta. "Error mitigation for short-depth quantum circuits". *Physical Review Letters* 119 (2017), p. 180509. DOI: 10.1103/PhysRevLett.119.180509.

[72] Ying Li and Simon C. Benjamin. "Efficient Variational Quantum Simulator Incorporating Active Error Minimization". *Physical Review X* 7 (2 2017), p. 021050. DOI: 10.1103/PhysRevX.7.021050.

[73] Yihui Quek et al. "Exponentially tighter bounds on limitations of quantum error mitigation". *arXiv:2210.11505* (2022). DOI: 10.48550/arXiv.2210.11505.

[74] Ryuji Takagi, Hiroyasu Tajima, and Mile Gu. "Universal sampling lower bounds for quantum error mitigation". *arXiv:2208.09178* (2022). DOI: 10.1103/PhysRevLett.131.210602.

[75] Joel J. Wallman and Joseph Emerson. "Noise tailoring for scalable quantum computation via randomized compiling". *Physical Review A* 94 (5 2016), p. 052325. DOI: 10.1103/PhysRevA.94.052325.

[76] Ewout van den Berg, Zlatko K. Minev, and Kristan Temme. "Model-free readout-error mitigation for quantum expectation values". *Phys. Rev. A* 105 (3 2022), p. 032620. DOI: 10.1103/PhysRevA.105.032620.

[77] Alexander Erhard et al. "Characterizing large-scale quantum computers via cycle benchmarking". *Nature Communications* 10.1 (Nov. 2019). DOI: 10.1038/s41467-019-13068-7.

[78] Senrui Chen et al. "Quantum advantages for Pauli channel estimation". *Physical Review A* 105.3 (Mar. 2022). DOI: 10.1103/physreva.105.032435.

[79] Steven T. Flammia and Joel J. Wallman. "Efficient Estimation of Pauli Channels". *ACM Transactions on Quantum Computing* 1.1 (2020), pp. 1–32. DOI: 10.1145/3408039.

[80] William J. Huggins et al. "Virtual Distillation for Quantum Error Mitigation". *Physical Review X* 11.4 (Nov. 2021). DOI: 10.1103/physrevx.11.041036.

[81] Bálint Koczor. "Exponential Error Suppression for Near-Term Quantum Devices". *Physical Review X* 11.3 (Sept. 2021). DOI: 10.1103/physrevx.11.031057.

[82] Changjun Kim, Kyungdeock Daniel Park, and June-Koo Rhee. "Quantum Error Mitigation With Artificial Neural Network". *IEEE Access* 8 (2020), pp. 188853–188860. DOI: 10.1109/ACCESS.2020.3031607.

[83] Piotr Czarnik et al. "Error mitigation with Clifford quantum-circuit data". *Quantum* 5 (2021), p. 592. DOI: 10.22331/Q-2021-11-26-592.

[84] Piotr Czarnik et al. "Improving the efficiency of learning-based error mitigation". *arXiv:2204.07109* (2022). DOI: 10.48550/arXiv.2204.07109.

[85] Elizabeth R. Bennewitz et al. "Neural Error Mitigation of Near-Term Quantum Simulations". *Nature Machine Intelligence* 4 (7 2022), pp. 618–624. DOI: 10.1038/s42256-022-00509-0.

[86] Tirthak Patel and Devesh Tiwari. "Qraft: Reverse Your Quantum Circuit and Know the Correct Program Output". *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. 2021, pp. 443–455. DOI: 10.1145/3445814.3446743.

[87] Armands Strikis et al. "Learning-Based Quantum Error Mitigation". *PRX Quantum* 2 (4 2021), p. 040330. DOI: 10.1103/PRXQuantum.2.040330.

[88] Hsin-Yuan Huang et al. "Power of data in quantum machine learning". *Nature Communications* 12.1 (2021). DOI: 10.1038/s41467-021-22539-9.

[89] Hsin-Yuan Huang et al. "Provably efficient machine learning for quantum many-body problems". *Science* 377.6613 (2022). DOI: 10.1126/science.abk3333.

[90] Pedro Rivero et al. *Zero Noise Extrapolation prototype*. https://github.com/qiskit-community/prototype-zne. 2022.

[91] Ivan Henao, Jader P. Santos, and Raam Uzdin. "Adaptive quantum error mitigation using pulse-based inverse evolutions". *arXiv:2303.05001* (2023). DOI: 10.1038/s41534-023-00785-7.

[92] Nic Ezzell et al. "Dynamical decoupling for superconducting qubits: a performance survey". *arXiv:2207.03670* (2022). DOI: 10.48550/arXiv.2207.03670.

[93] Bibek Pokharel and Daniel A. Lidar. "Demonstration of algorithmic quantum speedup". *arXiv:2207.07647* (2022). DOI: 10.1103/physrevlett.130.210602.

[94] Aram W. Harrow and Richard A. Low. "Random Quantum Circuits are Approximate 2-designs". *Communications in Mathematical Physics* 291.1 (2009), pp. 257–302. DOI: 10.1007/s00220-009-0873-6.

[95] Haoran Liao et al. "Robust in practice: Adversarial attacks on quantum machine learning". *Physical Review A* 103 (4 2021), p. 042427. DOI: 10.1103/PhysRevA.103.042427.

[96] Oles Shtanko et al. "Uncovering Local Integrability in Quantum Many-Body Dynamics". *arXiv:2307.07552* (2023). DOI: 10.48550/arXiv.2307.07552.

[97] Abhinav Kandala et al. "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets". *Nature* 549.7671 (2017), pp. 242–246. DOI: 10.1038/nature23879.

[98] Rex Ying et al. "Graph Convolutional Neural Networks for Web-Scale Recommender Systems". *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. London, United Kingdom, 2018, pp. 974–983. DOI: 10.1145/3219819.3219890.

[99] Patrick Reiser et al. "Graph neural networks for materials science and chemistry". *Communications Materials* 3.1 (93 2022). DOI: 10.1038/s43246-022-00315-6.

[100] Yunsheng Shi et al. "Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification". *Proceedings of the 13th International Joint Conference on Artificial Intelligence, IJCAI-21*. Aug. 2021, pp. 1548–1554. DOI: 10.24963/ijcai.2021/214.

[101] Ekagra Ranjan, Soumya Sanyal, and Partha Pratim Talukdar. "ASAP: Adaptive Structure Aware Pooling for Learning Hierarchical Graph Representations". *AAAI Conference on Artificial Intelligence*. 2019. DOI: 10.48550/arXiv.1911.07979.

[102] Daniel Bultrini et al. "Unifying and benchmarking state-of-the-art quantum error mitigation techniques". *Quantum* 7 (2023), p. 1034. DOI: 10.22331/q-2023-06-06-1034.

[103] Jacob Biamonte et al. "Quantum machine learning". *Nature* 549.7671 (2017), pp. 195–202. DOI: 10.1038/nature23474.

[104] John Preskill. "Quantum Computing in the NISQ Era and Beyond". *Quantum* 2 (2018), p. 79. DOI: 10.22331/q-2018-08-06-79.

[105] Yi Xia et al. "Quantum-enhanced Data Classification with a Variational Entangled Sensor Network". *arXiv: 2006.11962* (2020). DOI: 10.1103/PhysRevX.11.021047.

[106] Christian Szegedy et al. "Intriguing Properties of Neural Networks". *ICLR*. 2014. DOI: 10.48550/arXiv.1312.6199.

[107] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and Harnessing Adversarial Examples". *ICLR*. 2015. DOI: 10.48550/arXiv.1412.6572.

[108] Nilesh Dalvi et al. "Adversarial Classification". *ACM*. 2004, p. 99. DOI: 10.1145/1014052.1014066.

[109] Mesut Ozdag. "Adversarial Attacks and Defences: A Survey". *Procedia Computer Science* 140 (2018), pp. 152–161.

[110] Battista Biggio and Fabio Roli. "Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning". *Pattern Recognition* 84 (2018), pp. 317–331. DOI: 10.1016/j.patcog.2018.07.023.

[111] Ling Huang et al. "Adversarial Machine Learning". *ACM*. 2011, pp. 43–57. DOI: 10.1145/2046684.2046692.

[112] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. "Robustness of classifiers: from adversarial to random noise". *NIPS*. 2016, pp. 1632–1640. DOI: 10.48550/arXiv.1608.08967.

[113] Saeed Mahloujifar, Dimitrios I. Diochnos, and Mohammad Mahmoody. "The Curse of Concentration in Robust Learning: Evasion and Poisoning Attacks from Concentration of Measure". *AAAI*. Vol. 33. 2019, pp. 4536–4543. DOI: https://doi.org/10.1609/aaai.v33i01.33014536.

[114] Justin Gilmer et al. "The Relationship Between High-Dimensional Geometry and Adversarial Examples". *arXiv:1801.02774v3* (2018).

[115] Dimitrios I. Diochnos, Saeed Mahloujifar, and Mohammad Mahmoody. "Adversarial Risk and Robustness: General Definitions and Implications for the Uniform Distribution". *NIPS*. 2018, pp. 10380–10389. DOI: 10.48550/arXiv.1810.12272.

[116] Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. "Analysis of Classifiers' Robustness to Adversarial Perturbations". *Machine Learning* 107 (2018), pp. 481–508. DOI: 10.48550/arXiv.1502.02590.

[117] Alhussein Fawzi, Hamza Fawzi, and Omar Fawzi. "Adversarial Vulnerability for Any Classifier". *NIPS*. 2018, pp. 1186–1195. DOI: 10.48550/arXiv.1802.08686.

[118] Nana Liu and Peter Wittek. "Vulnerability of Quantum Classification to Adversarial Perturbations". *Physical Review A* 101.062331 (2020). DOI: 10.1103/PhysRevA.101.062331.

[119] Sirui Lu, Lu-Ming Duan, and Dong-Ling Deng. "Quantum Adversarial Machine Learning". *Physical Review Research* 2.033212 (2020). DOI: 10.1103/PhysRevResearch.2.033212.

[120] Yuxuan Du et al. "Quantum Noise Protects Quantum Classifiers Against Adversaries". *arXiv: 2003.09416* (2020). DOI: 10.1103/PhysRevResearch.3.023153.

[121] Ji Guan, Wang Fang, and Mingsheng Ying. "Robustness Verification of Quantum Machine Learning". *arXiv:2008.07230* (2020). DOI: 10.1007/978-3-030-81685-8_7.

[122] Michel Ledoux. *The Concentration of Measure Phenomenon*. Vol. 89. Mathematical Surveys and Monographs. American Mathematical Society, 2001.

[123] Vitali D. Milman, Gideon Schechtman. *Asymptotic Theory of Finite Dimensional Normed Spaces: Isoperimetric Inequalities in Riemannian Manifolds*. Lecture Notes in Mathematics. Springer, 2002. DOI: 10.1007/978-3-540-38822-7.

[124] Jarrod R. McClean et al. "Barren Plateaus in Quantum Neural Network Training Landscapes". *Nature Communications* 9.1 (2018). DOI: 10.1038/s41467-018-07090-4.

[125] Sandu Popescu, Anthony J. Short, and Andreas Winter. "Entanglement and the Foundations of Statistical Mechanics". *Nature Physics* 2.11 (2006), pp. 754–758. DOI: 10.1038/nphys444.

[126] Markus P. Müller, David Gross, and Jens Eisert. "Concentration of Measure for Quantum States with a Fixed Expectation Value". *Communications in Mathematical Physics* 303.3 (2011), pp. 785–824. DOI: 10.1007/s00220-011-1205-1.

[127] Danilo Jimenez Rezende and Shakir Mohamed. "Variational Inference with Normalizing Flows". *PMLR*. Vol. 37. 2015, pp. 1530–1538.

[128] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". *ICLR*. 2014. DOI: 10.48550/arXiv.1312.6114.

[129] Ian J Goodfellow et al. "Generative Adversarial Nets". *NIPS*. 2014, pp. 2672–2680.

[130] Piotr Bojanowski et al. "Optimizing the Latent Space of Generative Networks". *PMLR*. Vol. 80. 2018, pp. 600–609. DOI: 10.48550/arXiv.1707.05776.

[131] Martin Arjovsky, Soumith Chintala, and Leon Bottou. "Wasserstein Generative Adversarial Networks". *PMLR*. Vol. 70. 2017, pp. 214–223.

[132] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. "Adversarial Machine Learning at Scale". *ICLR*. 2017. DOI: 10.48550/arXiv.1611.01236.

[133] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. "Adversarial Examples in the Physical World". *ICLR*. 2019. DOI: 10.48550/arXiv.1607.02533.

[134] Sébastien Bubeck, Eric Price, and Ilya Razenshteyn. "Adversarial Examples from Computational Constraints". *PMLR*. Vol. 97. 2019, pp. 831–840. DOI: 10.48550/arXiv.1805.10204.

[135] Zachary Charles, Harrison Rosenberg, and Dimitris Papailiopoulos. "A Geometric Perspective on the Transferability of Adversarial Directions". *PMLR*. Vol. 89. 2018, pp. 1960–1968. DOI: 10.48550/arXiv.1811.03531.

[136] Logan Engstrom et al. "Adversarial Examples Are Not Bugs, They Are Features". *NIPS*. 2019, pp. 125–136. DOI: 10.48550/arXiv.1905.02175.

[137] Jan Philip Göpfert et al. "Adversarial Attacks Hidden in Plain Sight". *Advances in Intelligent Data Analysis* XVIII (2020), pp. 235–247. DOI: 10.1007/978-3-030-30487-4_24.

[138] William Huggins et al. "Towards Quantum Machine Learning with Tensor Networks". *Quantum Science and Technology* 4.2 (2019), p. 24001. DOI: 10.1088/2058-9565/aaea94.

[139] Marcello Benedetti et al. "Parameterized Quantum Circuits as Machine Learning Models". *Quantum Science and Technology* 4.4 (2019), p. 043001. DOI: 10.1088/2058-9565/ab4eb5.

[140] E. Miles Stoudenmire and David J. Schwab. "Supervised Learning with Quantum-Inspired Tensor Networks". *NIPS*. 2016, pp. 4799–4807. DOI: 10.48550/arXiv.1605.05775.

[141] Edward Grant et al. "Hierarchical Quantum Classifiers". *Quantum Information* 4.1 (2018), p. 65. DOI: 10.1038/s41534-018-0116-9.

[142] Shuxiang Cao et al. "Cost-function Embedding and Dataset Encoding for Machine Learning with Parameterized Quantum Circuits". *Physical Review A* 101.5 (2020), p. 052309. DOI: 10.1103/PhysRevA.101.052309.

[143] John Martyn et al. "Entanglement and Tensor Networks for Supervised Image Classification". *arXiv:2007.06082* (2020). DOI: 10.48550/arXiv.2007.06082.

[144] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. "Quantum Random Access Memory". *Physical Review Letters* 100.16 (2008), p. 160501. DOI: 10.1103/PhysRevLett.100.160501.

[145] Ryan Larose and Brian Coyle. "Robust Data Encodings for Quantum Classifiers". *Physical Review A* 102.3 (2020), p. 032420. DOI: 10.1103/PhysRevA.102.032420.

[146] Roman Vershynin. "Four lectures on probabilistic methods for data science". *arXiv:1612.06661* (2017). DOI: 10.48550/arXiv.1612.06661.

[147] Christer Borell. "The Brunn-Minkowski inequality in Gauss space". *Inventiones Mathematicae* 30.2 (1975), pp. 207–216. DOI: 10.1007/BF01425510.

[148] M. Gromov and V. D. Milman. "A Topological Application of the Isoperimetric Inequality". *American Journal of Mathematics* 105.4 (1983), pp. 843–854. DOI: 10.2307/2374298.

[149] Thierry Giordano and Vladimir Pestov. "Some Extremely Amenable Groups Related to Operator Algebras and Ergodic Theory". *Journal of the Institute of Mathematics of Jussieu* 6 (2007). DOI: 10.48550/arXiv.math/0405288.

[150] Gamaleldin F. Elsayed et al. "Adversarial Examples That Fool Both Computer Vision and Time-limited Humans". *NIPS*. 2018, pp. 3910–3920. DOI: 10.48550/arXiv.1802.08195.

[151] Robert Lockhart. "Low-rank Separable States Are A Set of Measure Zero Within the Set of Low-rank States". *Physical Review A* 65.6 (2002), p. 064304. DOI: 10.1103/PhysRevA.65.064304.

[152] Ajil Jalal, Andrew Ilyas, and Constantinos Daskalakis. "The Robust Manifold Defense: Adversarial Training using Generative Models". *arXiv: 1712.09196v5* (2019). DOI: 10.48550/arXiv.1712.09196.

[153] Mahyar Khayatkhoei, Maneesh K. Singh, and Ahmed Elgammal. "Disconnected Manifold Learning for Generative Adversarial Networks". *NIPS*. 2018, pp. 7343–7353. DOI: 10.48550/arXiv.1806.00880.

[154] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". *ICLR*. 2016. DOI: 10.48550/arXiv.1511.06434.

[155] Jens Behrmann et al. "Invertible Residual Networks". *PMLR*. Vol. 97. 2019, pp. 573–582. DOI: 10.48550/arXiv.1811.00995.

[156] Han Zhang et al. "Self-Attention Generative Adversarial Networks". *PMLR*. Vol. 97. 2019, pp. 7354–7363. DOI: 10.48550/arXiv.1805.08318.

[157] Takeru Miyato et al. "Spectral Normalization for Generative Adversarial Networks". *ICLR*. 2018. DOI: 10.48550/arXiv.1802.05957.

[158] Zhengli Zhao, Dheeru Dua, and Sameer Singh. "Generating Natural Adversarial Examples". *ICLR*. 2018. DOI: 10.48550/arXiv.1710.11342.

[159] Patrick J. Coles, M. Cerezo, and Lukasz Cincio. "Strong bound between trace distance and Hilbert-Schmidt distance for low-rank states". *Physical Review A* 100.2 (2019), p. 022103. DOI: 10.1103/PhysRevA.100.022103.

[160] D Spehner et al. "Geometric Measures of Quantum Correlations with Bures and Hellinger Distances". *Lectures on General Quantum Correlations and their Applications* (2017), pp. 105–157. DOI: 10.48550/arXiv.1611.03449.

[161] Alexander S. Holevo. "On Quasiequivalence of Locally Normal States". *Theoretical and Mathematical Physics* 13 (1972), pp. 1071–1082.

[162] Haoran Liao et al. "Decohering tensor network quantum machine learning models". *Quantum Machine Intelligence* 5.1 (2023). DOI: 10.1007/s42484-022-00095-9.

[163] Roman Orus and Rolf Tarrach. "Weakly-entangled States are Dense and Robust". *Physical Review A* 70.5 (2004), p. 050101. DOI: 10.1103/PhysRevA.70.050101.

[164] Jacob Miller, Geoffrey Roeder, and Tai-Danae Bradley. "Probabilistic Graphical Models and Tensor Networks: A Hybrid Framework". *arXiv:2106.15666* (2021). DOI: 10.48550/arXiv.2106.15666.

[165] Jacob Biamonte and Ville Bergholm. "Tensor Networks in a Nutshell". *arXiv:1708.00006* (2017). DOI: 10.48550/arXiv.1708.00006.

[166] Glen Evenbly and Guifré Vidal. "Tensor Network States and Geometry". *Journal of Statistical Physics* 145.4 (2011), pp. 891–918. DOI: 10.1007/s10955-011-0237-4.

[167] Jens Eisert. "Entanglement and Tensor Network States". *Emergent Phenomena in Correlated Matter Modeling and Simulation*. Ed. by Eva Pavarini, Erik Koch, and Ulrich Schollwöck. Vol. 3. Verlag des Forschungszentrum Jülich, 2013. Chap. 17. DOI: 10.48550/arXiv.1308.3318.

[168] Ian Convy et al. "Mutual information scaling for tensor network machine learning". *Machine Learning: Science and Technology* 3.1 (2022), p. 015017. DOI: 10.1088/2632-2153/ac44a9.

[169] Sirui Lu et al. "Tensor networks and efficient descriptions of classical data". *arXiv:2103.06872* (2021). DOI: 10.48550/arXiv.2103.06872.

[170] Don N. Page. "Average entropy of a subsystem". *Physical Review Letters* 71.9 (1993), pp. 1291–1294. DOI: 10.1103/physrevlett.71.1291.

[171] M B Hastings. "An area law for one-dimensional quantum systems". *Journal of Statistical Mechanics: Theory and Experiment* 2007.08 (2007), P08024–P08024. DOI: 10.1088/1742-5468/2007/08/p08024.

[172] M. Cramer et al. "Entanglement-area law for general bosonic harmonic lattice systems". *Phys. Rev. A* 73 (1 2006), p. 012309. DOI: 10.1103/PhysRevA.73.012309.

[173] J. Eisert, M. Cramer, and M. B. Plenio. "Colloquium: Area laws for the entanglement entropy". *Rev. Mod. Phys.* 82 (1 2010), pp. 277–306. DOI: 10.1103/RevModPhys.82.277.

[174] G. Vidal et al. "Entanglement in Quantum Critical Phenomena". *Phys. Rev. Lett.* 90 (22 2003), p. 227902. DOI: 10.1103/PhysRevLett.90.227902.

[175] G. Evenbly and G. Vidal. "Algorithms for entanglement renormalization". *Physical Review B - Condensed Matter and Materials Physics* 79.14 (2009). arXiv: 0707.1454. DOI: 10.1103/PhysRevB.79.144108.

[176] Jacob C. Bridgeman and Christopher T. Chubb. "Hand-waving and interpretive dance: An introductory course on tensor networks". *Journal of Physics A: Mathematical and Theoretical* 50.22 (2017). DOI: 10.1088/1751-8121/aa6dc3.

[177] Yoav Levine et al. "Deep Learning and Quantum Entanglement: Fundamental Connections with Implications to Network Design". *Proceedings of ICLR*. 2018. DOI: 10.48550/arXiv.1704.01552.

[178] Nadav Cohen and Amnon Shashua. "Convolutional Rectifier Networks as Generalized Tensor Decompositions". *Proceedings of ICML*. 2016, pp. 955–963. DOI: 10.48550/arXiv.1603.00162.

[179] Yaoyun Shi, Luming Duan, and Guifré Vidal. "Classical simulation of quantum many-body systems with a tree tensor network". *Physical Review A* 74 (2 2006), p. 022320. DOI: 10.1103/PhysRevA.74.022320.

[180] Guifré Vidal. "Entanglement renormalization". *Physical Review Letters* 99 (22 2007), pp. 1–4. DOI: 10.1103/PhysRevLett.99.220405.

[181] E. Miles Stoudenmire. "Learning relevant features of data with multi-scale tensor networks". *Quantum Science and Technology* 3.3 (2018), p. 034003. DOI: 10.1088/2058-9565/aaba1a.

[182] Justin A. Reyes and E. Miles Stoudenmire. "Multi-scale tensor network architecture for machine learning". *Machine Learning: Science and Technology* 2.3 (2021), p. 035036. DOI: 10.1088/2632-2153/abffe8.

[183] Michael L. Wall and Giuseppe D'Aguanno. "Tree-tensor-network classifiers for machine learning: From quantum inspired to quantum assisted". *Physical Review A* 104.4 (2021), p. 042408. DOI: 10.1103/PhysRevA.104.042408.

[184] Edward Grant et al. "Hierarchical quantum classifiers". *npj Quantum Information* 4.1 (2018), p. 65. DOI: 10.1038/s41534-018-0116-9.

[185] William J. Huggins et al. "Towards quantum machine learning with tensor networks". *Quantum Science and Technology* 4.2 (2019), p. 024001. DOI: 10.1088/2058-9565/aaea94.

[186] Iris Cong, Soonwon Choi, and Mikhail D. Lukin. "Quantum convolutional neural networks". *Nature Physics* 15.12 (2019), pp. 1273–1278. DOI: 10.1038/s41567-019-0648-8.

[187] Kosuke Mitarai et al. "Quantum Circuit Learning". *Physical Review A* 98.3 (2018), p. 032309. DOI: 10.1103/PhysRevA.98.032309.

[188] Marcello Benedetti et al. "Parameterized Quantum Circuits as Machine Learning Models". *Quantum Science and Technology* 4.4 (2019), p. 043001. DOI: 10.1088/2058-9565/ab4eb5.

[189] Vojtěch Havlíček et al. "Supervised learning with quantum-enhanced feature spaces". *Nature* 567.7747 (2019), pp. 209–212. DOI: 10.1038/s41586-019-0980-2.

[190] Elina Robeva and Anna Seigal. "Duality of Graphical Models and Tensor Networks". *Information and Inference* 8.2 (2019), pp. 273–88. DOI: 10.48550/arXiv.1710.01437.

[191] Ivan Glasser et al. "Expressive power of tensor-network factorizations for probabilistic modeling, with applications from hidden Markov models to quantum machine learning". *Proceedings of NIPS*. 2019, pp. 1498–1510. DOI: 10.48550/arXiv.1907.03741.

[192] E. Miles Stoudenmire and David J. Schwab. "Supervised Learning with Quantum-Inspired Tensor Networks". *Proceedings of NIPS*. 2016, pp. 4799–4807. DOI: 10.48550/arXiv.1605.05775.

[193] Ryan Larose and Brian Coyle. "Robust Data Encodings for Quantum Classifiers". *Physical Review A* 102.3 (2020), p. 032420. DOI: 10.1103/PhysRevA.102.032420.

[194] Gregor Tanner. "Unitary-stochastic matrix ensembles and spectral statistics". *Journal of Physics A: Mathematical and General* 34.41 (2001), pp. 8485–8500. DOI: 10.1088/0305-4470/34/41/307.

[195] Karol Zyczkowski et al. "Random unistochastic matrices". *Journal of Physics A: Mathematical and General* 36.12 (2003), pp. 3425–3450. DOI: 10.1088/0305-4470/36/12/333.

[196] John Watrous. *The Theory of Quantum Information*. Cambridge University Press, 2018.

[197] Kaiming He et al. "Deep Residual Learning for Image Recognition". *Proceedings of CVPR*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[198] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. "MNIST handwritten digit database". *http://yann.lecun.com/exdb/mnist* (2010).

[199] Tarin Clanuwat et al. "Deep Learning for Classical Japanese Literature". *arXiv:1812.01718* (2018). DOI: 10.20676/00000341.

[200] Han Xiao, Kashif Rasul, and Roland Vollgraf. "Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms". *arXiv:1708.07747* (2017). DOI: 10.48550/arXiv.1708.07747.