UNIVERSITY OF SOUTHAMPTON

Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

# Quantum Error Mitigation for Error-Resilient Quantum Computation

*by*

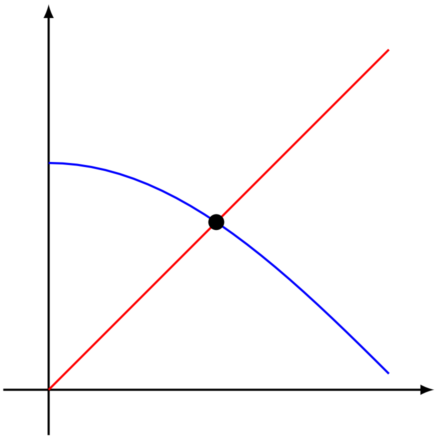
**Yifeng Xiong**
BSc, MSc
ORCiD: 0000-0002-4290-7116

*A thesis submitted in partial fulfillment of the*
*requirements for the award of Doctor of Philosophy*
*at University of Southampton*

June 2022

SUPERVISORS:
**Prof. Lajos Hanzo**
FREng, FIEEE, FIET, DSc
Chair of Next Generation Wireless Research Group
and
**Prof. Soon Xin Ng**
PhD, BEng, CEng, FIET, SMIEEE, FHEA

*Dedicated to my super Cosine,*
*for your* $\operatorname{supp}(\cos)$ *satisfying* $\frac{\partial}{\partial x}\operatorname{supp}(\cos) = 0$ *and*
$|\operatorname{supp}(\cos)| = \aleph_1$*, where* $\operatorname{supp}(\cos) := \{x \in \mathbb{R} | \cos(x) \neq 0\}$



University of Southampton

Abstract

Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

Doctor of Philosophy

**Quantum Error Mitigation for Error-Resilient Quantum Computation**

by Yifeng Xiong

Quantum computers have the potential of providing unprecedented computational power for solving problems that are known to be difficult for classical computers, including integer factoring, database searching, molecular simulation and combinatorial optimization. At the time of writing, the largest real-world quantum computers are only handling 100–200 noisy qubits, which are also known as "noisy intermediate-scale quantum computers".

Quantum error mitigation (QEM) constitutes a class of promising techniques, which are capable of reducing the computational error caused by the decoherence-induced impairments in the qubits. However, this is achieved at the cost of an undesired computational overhead termed as the sampling overhead. In this thesis, we aim for striking a beneficial and flexible computational accuracy vs. sampling overhead trade-off, for both circuit-level and algorithm-level QEM. We commence by presenting an overview of existing QEM techniques, highlighting their main sources of computational overhead and their requirements concerning the prior knowledge about the computational task or the noise model.

Specifically, for *channel-inversion based QEM*, we present a comprehensive **sampling overhead analysis**, and propose "quantum channel precoders" that reduce the sampling overhead. We show that Pauli channels have the lowest sampling overhead in a large class of practical quantum channels, and that the depolarizing channels have the lowest sampling overhead among Pauli channels. Furthermore, we conceive a beneficial amalgam of channel-inversion based QEM and quantum error-correction codes (QECCs).

Regarding the specific implementation strategy of channel-inversion based QEM, we analyze the error scaling versus sampling overhead tradeoff of the **Monte Carlo based channel inversion**, which drastically reduces the complexity of candidate circuit generation, at the cost of a moderate accuracy degradation. In particular, we show that the computational error of the Monte Carlo based channel inversion is on the order of $O(\sqrt{N_{\mathrm{G}}})$, where $N_{\mathrm{G}}$ is the number of gates. This is similar to that of the exact channel

inversion. By contrast, the computational error is on the order of $O(N_{\mathrm{G}})$ when no QEM is applied. However, the candidate circuit generation complexity of the Monte Carlo based strategy is exponentially lower than that of the exact channel inversion, implying that the Monte Carlo based channel inversion strategy has a favorable accuracy vs. overhead trade-off.

Next, we design *symmetry-based QEM* methods having high sample-efficiency. In general, the intrinsic symmetry conditions regarding the computational tasks may be exploited for the mitigation of the computational error using the method of symmetry verification. However, it is limited to state symmetries, hence it has a restricted scope of application. We extend the symmetry verification method to circuit symmetries by proposing the technique of **spatio-temporal stabilizers**. Specifically, as a natural generalization of the conventional stabilizers, spatio-temporal stabilizers are capable of detecting whether a quantum circuit commutes with certain operators. This enables us to mitigate the errors violating the commutativity conditions. We then discuss the detailed design strategy of the spatio-temporal stabilizer method for mitigating the errors of practical quantum algorithms, including the quantum Fourier transform and the quantum approximate optimization algorithm.

Regarding a specific kind of symmetry conditions, namely the permutation symmetry across different activations of quantum circuits, we propose the method of **permutation filtering**, inspired by the philosophy of finite impulse response (FIR) filters of classical signal processing theory. Remarkably, the existing virtual distillation method may be viewed as a special case of permutation filters. We show that the proposed design of these filters converges to the global optimum, and that the error reduction performance of the optimal filter is particularly good for narrowband noises, corresponding to the scenario of deep quantum circuits.

Concerning the spectral leakage issue of the quantum phase estimation algorithm, we propose an *algorithm-level error mitigation* method, namely the **dual-frequency estimator**. This is potentially useful in the context of noisy intermediate-scale quantum computing, since the maximum achievable recording length $N$ is ultimately restricted by the coherence time, hence we have to rely on multiple samples when a high phase estimation accuracy is required. In particular, we show that when the number of samples is sufficiently large, the dual-frequency estimator outperforms the cosine window, which is shown to be optimal for single-sample estimation. Furthermore, we also show that the dual-frequency estimator achieves the Cramér-Rao bound when the number of samples is large. This implies that the dual-frequency estimator has a beneficial accuracy vs. sampling overhead trade-off in the asymptotic regime.

# Declaration of Authorship

I declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published in [1–4], and submitted for potential future publication in [5].

Signed:......................................................................... Date:..................

# Acknowledgements

I would like to express my sincere gratitude to my supervisors, Prof. Lajos Hanzo and Prof. Soon Xin Ng for their great support and guidance during my journey across the quantum world. Their incredible and infinitely imaginative ideas inspired me to overcome the formidable challenges along the journey, and to achieve what seems to be impossible at the first glance.

I wish to thank Dr Daryus Chandra, Dr Chao Xu, Dr Zunaira Babar, Dr Rosie Cane, Dr Bálint Koczor, Prof. Gui-Lu Long, Dr Dong Pan, Dr Yun Chen, Dr Jingjing Cui, Dr Dong Liu, and Mr Xin Liu, for their inspirational suggestions and the fruitful discussions. I would also like to thank Prof. Lie-Liang Yang, Prof. Sheng Chen, Prof. Robert G. Maunder and Dr Mohammed El-Hajjar for their assistance.

I wish to acknowledge the use of IRIDIS High-Performance Computing Facility and the associated services at the University of Southampton in the completion of this work. I would also like to thank China Scholarship Council (CSC) for the financial support.

Thanks to all the lovely stars in this Spacetime and all parallel Spacetimes, and to the great unitary evolution of the entire Universe, which make this thesis finally collapse to its state of completion.

Finally, I wish to thank my parents, Jianming Xiong and Hong Lin, as well as my girlfriend Siqi Han, for their love, patience and support helped me thrive through the most difficult period of time.

# List of Publications

## Published Papers:

1. **Y. Xiong**, S. X. Ng, and L. Hanzo, "The accuracy vs. sampling overhead trade-off in quantum error mitigation using Monte Carlo-based channel inversion," *IEEE Transactions on Communications*, vol. 70, no. 3, pp. 1943–1956, Mar. 2022.
2. **Y. Xiong**, S. X. Ng, and L. Hanzo, "Quantum error mitigation relying on permutation filtering," *IEEE Transactions on Communications*, vol. 70, no. 3, pp. 1927–1942, Mar. 2022.
3. **Y. Xiong**, D. Chandra, S. X. Ng, and L. Hanzo, "Sampling overhead analysis of quantum error mitigation: Uncoded vs. coded systems," *IEEE Access*, vol. 8, pp. 228 967–228 991, 2020.
4. **Y. Xiong**, N. Wu, Y. Shen and M. Z. Win, "Cooperative Localization in Massive Networks," *IEEE Transactions on Information Theory*, vol. 68, no. 2, pp. 1237-1258, Feb. 2022.
5. **Y. Xiong**, S. X. Ng, G.-L. Long, and L. Hanzo, "Dual-Frequency Quantum Phase Estimation Mitigates the Spectral Leakage of Quantum Algorithms," *IEEE Signal Processing Letters*, vol. 29, pp. 1222–1226, Apr. 2022.
6. J. Cui, **Y. Xiong**, S. X. Ng, and L. Hanzo, "Quantum Approximate Optimization Algorithm Based Maximum Likelihood Detection," *IEEE Transations on Communications*, *Accepted*, 2022.
7. C. Xu, **Y. Xiong**, N. Ishikawa, R. Rajashekar, S. Sugiura, Z. Wang, S. X. Ng, L.-L. Yang, and L. Hanzo, "Space-, Time- and Frequency-Domain Index Modulation for Next-Generation Wireless: A Unified Single-/Multi-Carrier and Single-/Multi-RF MIMO Framework," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3847-3864, Jun. 2021.
8. Y. Chen, **Y. Xiong**, D. Chen, T. Jiang, S. X. Ng and L. Hanzo, "Hybrid Precoding for WideBand Millimeter Wave MIMO Systems in the Face of Beam Squint," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1847-1860, Mar. 2021.

## Submitted Papers for Potential Future Publication:

1. **Y. Xiong**, D. Chandra, S. X. Ng, and L. Hanzo, "Circuit-Oriented Symmetry Verification: From Quantum Switches to Spatio-Temporal Stabilizers," Submitted to *IEEE Transactions on Signal Processing*, 2021. [Online]. Available: `https://arxiv.org/abs/2112.13904`.
2. M. Zhang, **Y. Xiong**, S. X. Ng, and M. El-Hajjar, "Deployment of Energy-Efficient Aerial Communication Platforms with Low-Complexity Detection," Submitted to *IEEE Transactions on Vehicular Technology*, 2022.

# Contents

# Glossary

## List of Acronyms

**CI-QEM** channel-inversion quantum error mitigation

**CPTnI** completely positive trace-nonincreasing

**CPTP** completely positive trace-preserving

**CRB** Cramér-Rao bound

**CSS** Calderbank-Shor-Steane

**FIR** finite impulse response

**GEP** gate error probability

**GGEP** generalized gate error probability

**KKT** Karush-Kuhn-Tucker

**MIMO** multiple-input mutliple-output

**MSE** mean squared error

**NISQ** noisy intermediate-scale quantum

**PCM** parity check matrix

**PTM** Pauli transfer matrix

**QAOA** quantum approximate optimization algorithm

**QCP** quantum channel precoder

**QECC** quantum error correction code

**QEDC** quantum error detection code

**QEM** quantum error mitigation

**QFT** quantum Fourier transform

**QPCA** quantum principal component analysis

**QPR** quasi-probability representation

**QSA** quantum search algorithm

**QSC** quantum stabilizer code

**RMSE** root mean squared error

**SOF** sampling overhead factor

**STS** spatio-temporal stabilizer

**VD** virtual distillation

**VQA** variational quantum algorithm

**VQE** variational quantum eigensolver

## List of Notations

- Deterministic scalars, vectors and matrices are represented by $x$, $\boldsymbol{x}$, and $\boldsymbol{X}$, respectively, whereas their random counterparts are denoted as $\mathsf{x}$, $\boldsymbol{\mathsf{x}}$, and $\boldsymbol{\mathsf{X}}$, respectively, with the exception that the density matrices of quantum states are represented by non-bold italic symbols, such as $\rho$.

- Deterministic sets, random sets, and operators are denoted as $\mathscr{X}$, $\mathsf{X}$, and $\mathcal{X}$, respectively.

- The notations $\mathbf{1}_n$, $\mathbf{0}_n$, $\mathbf{0}_{m\times n}$, and $\boldsymbol{I}_k$, represent the $n$-dimensional all-one vector, the $n$-dimensional all-zero vector, the $m \times n$ dimensional all-zero matrix, and the $k \times k$ identity matrix, respectively.

- The notation $\|\boldsymbol{x}\|_p$ represents the $\ell_p$-norm of vector $\boldsymbol{x}$, and the subscript may be omitted when $p = 2$. For matrices, $\|\boldsymbol{A}\|_p$ denotes the matrix norm induced by the corresponding $\ell_p$ vector norm.

- The notation $[\boldsymbol{A}]_{i,j}$ denotes the $(i, j)$-th entry of matrix $\boldsymbol{A}$. For a vector $\boldsymbol{x}$, $[\boldsymbol{x}]_i$ denotes its $i$-th element. The submatrix obtained by extracting the $i_1$-th to $i_2$-th rows and the $j_1$-th to $j_2$-th columns from $\boldsymbol{A}$ is denoted as $[\boldsymbol{A}]_{i_1:i_2,j_1:j_2}$. The notation $[\boldsymbol{A}]_{:,i}$ represents the $i$-th column of $\boldsymbol{A}$, and $[\boldsymbol{A}]_{i,:}$ denotes the $i$-th row, respectively.

- The notation $\mathrm{diag}(\cdot)$ denotes a diagonal matrix obtained by placing its argument on the main diagonal, and $\mathrm{mdiag}(\boldsymbol{A})$ denotes the matrix obtained by setting all entries in matrix $\boldsymbol{A}$ to zero apart from the main diagonal.

- The notation $\mathrm{vec}(\boldsymbol{A})$ denotes the vector obtained by vectorizing matrix $\boldsymbol{A}$, and $\mathrm{vec}^{-1}(\cdot)$ denotes the inverse operation.
- The complex conjugate of $\boldsymbol{A}$ is denoted as $\boldsymbol{A}^{\dagger}$. Similarly, the complex adjoint of an operator $\mathcal{X}$ is also denoted as $\mathcal{X}^{\dagger}$.
- The trace of a sqaure matrix $\boldsymbol{A}$ is denoted as $\mathrm{Tr}\{\boldsymbol{A}\}$.
- The notation $\boldsymbol{A} \otimes \boldsymbol{B}$ represents the Kronecker product between matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. The tensor product of operators $\mathcal{A}$ and $\mathcal{B}$ is also denoted as $\mathcal{A} \otimes \mathcal{B}$. Furthermore, the Cartesian product between sets $\mathscr{A}$ and $\mathscr{B}$ is denoted as $\mathscr{A} \otimes \mathscr{B}$ as well.
- The notation $\boldsymbol{A} \odot \boldsymbol{B}$ denotes the Hadamard product between matrices $\boldsymbol{A}$ and $\boldsymbol{B}$.
- Pure quantum states are denoted by "kets" $|\psi\rangle$, and their dual vectors are denoted by "bras" $\langle\psi|$.
- The notations $\mathbb{E}\{\cdot\}$, $\mathrm{Var}\{\cdot\}$, and $\mathrm{Cov}\{\cdot\}$ represent the expectation, the variance, and the covariance matrix of their arguments, respectively. The function $\mathbb{I}\{\cdot\}$ takes the value of one when its argument is a true expression, and takes the value of zero otherwise.
- The notation $1/\boldsymbol{x}$ represents the element-wise reciprocal of vector $\boldsymbol{x}$.
- The sign function $\mathrm{sgn}(x)$ is defined as

$$\mathrm{sgn}(x) = \begin{cases} 1, & x > 0; \\ 0, & x = 0; \\ -1, & x < 0. \end{cases}$$

- The Bachmann-Landau notations [6] used in this thesis are given as follows:

$$\begin{array}{ll} a(n) = O(b(n)) & \limsup_{n\to\infty} a(n)(b(n))^{-1} < \infty \\ a(n) = o(b(n)) & \liminf_{n\to\infty} a(n)(b(n))^{-1} = 0 \end{array}$$

# Chapter 1

# Introduction

## 1.1 Background and Motivations

On October 23rd, 2019, a group of scientists from Google declared for the first time "quantum supremacy": quantum computers can carry out certain tasks substantially faster than their best classical counterparts — supercomputers [7]. Specifically, Google's quantum computer Sycamore completed a task termed as Boson sampling [8] in 200 seconds, while a state-of-the-art classical supercomputer, IBM Summit, would take from 2.5 days (according to IBM [9]) to 10000 years (according to Google [7]).

Rather than a sudden and unexpected leap in computational power, quantum supremacy is enabled by the diligent and constant effort of the entire quantum computing community, spanning across a time period as long as 37 years. Indeed, the idea of quantum computation dates back to Feynman's seminal speech in 1982 [10], in which he suggested to build computers relying on quantum mechanics for the efficient simulation of quantum mechanical systems. Feynman's proposal was then condensed to a practical quantum algorithm, termed as Hamiltonian simulation [11], and also inspired a diverse range of other quantum algorithms providing speedup for applications far beyond the physics community [12–20]. Impressively, some quantum algorithms have shown to be capable of providing exponential speedup compared to the best classical algorithms [11, 12, 16].

The seemingly mysterious quantum speedup originates from the unique properties of quantum mechanics. Perhaps the most well-known metaphor illustrating these peculiar properties is the thought experiment of "Schrödinger's cat" [21]. In this treatise, we will save this poor cat from the miserable state of being simultaneously dead and alive, by presenting a more zoophilist-friendly example. Specifically, let us consider a coin in a black box. Classically, the coin either has its head or its tail side up, although we might not know which is true before we open the box. However, a "quantum coin" could stay in a coherent superposition of the “head state” and the “tail state” simultaneously. But once we open the box, the superposition is destroyed and we observe one of the two outcomes, where the probability of observing each outcome is determined by the way that the superposition is created. The action of “opening the box” is often referred to as “observation” or “measurement”. Let us now extend this example one step further, by considering a “box collection” containing $N$ such black boxes. Classical box collections would be in a specific state out of $2^N$ possible states, while quantum box collections could be in the superposition of all $2^N$ states. If we could properly manipulate the superposition state to create a desired outcome probability distribution, the quantum box collection works effectively as a quantum computer, in a way as if there were $2^N$ classical computers working in parallel. From this simple example, we see that quantum mechanical systems have the potential of providing exponential speedup due to the superposition-induced massive parallelization.

Although the achievement of quantum supremacy resulted in enormous excitement in the quantum computing society, it does not imply that ubiquitous quantum speedup is around the corner. In fact, the computational task of Boson sampling [8] chosen to demonstrate quantum supremacy in Google’s experiment has limited applications beyond the realms of computational complexity theory. Why quantum speedup has not been demonstrated on more popular and practical tasks? The short answer is: quantum computers produce errors. Moreover, the errors are difficult to correct, hence will accumulate as the computer executes its consecutive tasks, until the overall error rate becomes unacceptable at some stage. In particular, quantum computers suffer from the notorious quantum decoherence effects, which impose a perturbation on each and every quantum operation carried out by a “quantum gate”. Intuitively, the deleterious effect of decoherence may be viewed as exposing the quantum state to perpetual observation by the environment. As a quantum computer executes its program, the decoherence effect would gradually turn it into a classical computer, in the sense that it could only be in a single definitive state at a given time instance with high probability, instead of being in a superposition state. Ultimately, given a desired computational accuracy, decoherence sets a limit on the maximum duration of computation, termed as “coherence time” [22].

In principle, noisy quantum gates do not necessarily prevent quantum computation from being sufficiently accurate. A classical result, namely the threshold theorem [23], states that quantum computation may be carried out in the presence of decoherence with the help of quantum error correction codes (QECCs) [24–27], given that the error rate of each quantum gate is below a certain threshold. Generally speaking, QECCs protect a *logical* quantum bit (qubit) by mapping it to a larger set of *physical* qubits. The redundancy of the physical qubits ensures that errors perturbing a small fraction of the qubits can be detected and corrected with the help of some ancillary qubits (ancillas).

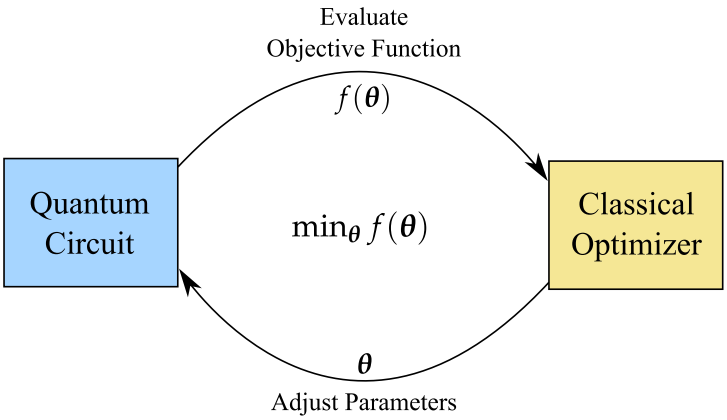


FIGURE 1.1: Illustration of the iterations in hybrid quantum-classical algorithms.

Moreover, the error-correction capability can be further enhanced by concatenating several QECCs, albeit naturally, at the expense of higher qubit overhead [28–30].

However, compared to the idealized quantum computing models considered in the threshold theorem, practical noisy intermediate-scale quantum computers may not be capable of supporting fully fault-tolerant operations, due to their limited number of qubits. Consequently, they may not be able to execute algorithms that require relatively long processing time, such as Shor's factorization algorithm [12] and Grover's quantum search algorithm [13]. Fortunately, there is evidence that algorithms tailored for noisy intermediate-scale quantum computers may yield superior performance compared to those of classical computers [31, 32]. Most of these algorithms belong to the category of hybrid quantum-classical algorithms, which exploit the power of classical computation to compensate for the short coherence time of quantum processors. As portrayed in Fig. 1.1, a typical hybrid quantum-classical algorithm would be performed in an iterative fashion. Due to the deep-rooted connections between the optimization problem and the variational method, hybrid quantum-classical algorithms are also referred to as "variational quantum algorithms" in the literature. The quantum circuit, which will be referred to as the function-evaluation circuit in this treatise, is designed to evaluate an objective function, given a set of input parameters [33]. The value of the objective function is then utilized in a classical optimizer, which computes an adjusted set of parameters for the next iteration. In general, the design of the function-evaluation circuit determines the application of the algorithm. Popular designs include the alternating operator circuit used in the quantum approximate optimization algorithm [32], and the "unitary coupled-cluster ansatz" circuit applied in the computation of molecular energy based on variational eigensolver [34].

Although the quantum circuits in hybrid quantum-classical algorithms have limited depth, their decoherence may still inflict non-negligible computational errors [35, 36]. This necessitates the design of low-qubit-overhead techniques for protecting quantum gates. As a design alternative to QECCs, the quantum error mitigation (QEM) philosophy has been proposed recently, which may correct the computational result without using a large number of ancillas [37–39]. The employment of quantum error mitigation (QEM) may facilitate error-resilient hybrid quantum-classical computation based on near-term quantum computers. In this treatise, we will introduce the design philosophy of QEM and its application to error-resilient quantum computation, followed by proposing novel QEM techniques aiming for striking a more favorable and flexible accuracy vs. overhead trade-off.

## 1.2 A Brief Overview of Error-Resilient Quantum Computation

### 1.2.1 Fault-Tolerant Quantum Computation

The research of error-resilient quantum computation has the ultimate goal of conceiving **fault-tolerant** quantum computing schemes. Broadly speaking, a quantum computer is said to be fault-tolerant, if its computational error rate does not degrade with time. The pursuit of fault-tolerance started from the investigation of quantum error correction codes (QECCs), which are designed for correcting local errors in quantum information processing systems.[1] The first QECC was invented by Shor [40], which encodes a single "logical qubit" into nine "physical qubits", and it is capable of protecting the physical qubits against any single-qubit error.

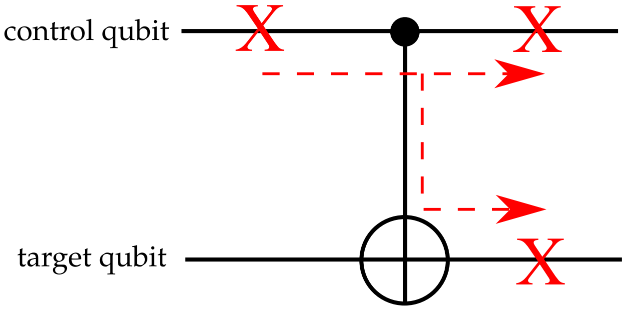


FIGURE 1.2: Illustration of the error proliferation phenomenon in a two-qubit quantum gate.

From the perspective of classical coding theory, Shor's code is not good enough in terms of its efficiency, since it only has a coding rate of $1/9$. Indeed, later investigations following this line of reasoning have arrived at QECCs having higher coding rates, including Steane's 7-qubit code [41] and Laflamme's 5-qubit code [42] having coding rates of $1/7$ and $1/5$, respectively. Laflamme's 5-qubit code is also referred to as the "perfect code", since it achieves certain upper bounds of the quantum coding rate,

---

[1] By saying "local errors", we mean the error patterns that involve only a small portion of the entire information processing system.

given a required minimum distance (in this case, 3 in order to correct a single-qubit error) [43].

To elaborate a little further, the minimum distance of a quantum code is defined as the minimum weight of the Pauli operator $\mathcal{E}$, so that for any pair of legitimate codewords $|i\rangle$ and $|j\rangle$, we have [24]

$$\langle i| \mathcal{E} |j\rangle = C_{\mathcal{E}} \mathbb{I}\{i = j\}, \tag{1.1}$$

where $C_{\mathcal{E}}$ is a coefficient depending only on the operator $\mathcal{E}$ (but not on the codeword indices $|i\rangle$ and $|j\rangle$). The weight of an operator $\mathcal{E}$ is defined as the number of qubits that $\mathcal{E}$ acts upon nontrivially (i.e. its action is not as if an identity gate) [24]. In other words, if a quantum code has a minimum distance of $d$, we see that any logical operation that turns a legitimate codeword into another would have a weight of at least $d$. Since the error operator $\mathcal{E}$ is a Pauli operator, one may employ the maximum likelihood decoding strategy, which applies another Pauli operator $\mathcal{R}$ to the perturbed state $\mathcal{E} |j\rangle$, so that $|i\rangle = \mathcal{R}\mathcal{E} |j\rangle$ is the nearest legitimate codeword to $\mathcal{E} |j\rangle$. Apparently, the maximum likelihood strategy will yield the correct state when the minimum distance of the quantum code satisfies $d \geqslant 2t + 1$, where $t$ is the weight of the error operator. This may be interpreted by stating that a quantum code a having minimum distance of $d = 2t + 1$ is capable of correcting $t$ errors. Naturally, one would expect a "good code" to have a high coding rate, while exhibiting a reasonable minimum distance for ensuring a high error correction capability.

However, the coding rate is not the only performance metric in the context of fault-tolerant quantum computation, and arguably not even the most important one. One of the main reasons is that computational systems suffer from the **error proliferation** phenomenon, which constitutes a major distinction from communication systems. To elaborate, let us consider the simple example portrayed in Fig. 1.2. The two-qubit gate shown in Fig. 1.2 is known as the CNOT gate, which flips the state of the target qubit, conditioned on the state of the control qubit. Observe that a pre-existing X-error (also known as a bit-flip error) entering the CNOT gate would perturb both qubits at the output of the CNOT gate. In general, all multi-qubit gates would induce error proliferation, hence the overall error rate would increase exponentially with the execution time of the quantum circuit. In light of this, error proliferation has to be circumvented in order for any QECC to be practically useful.

A simple but effective method of preventing error proliferation is to use **transversal gates** [44,45]. Consider a quantum register consisting of $m$ distinct and non-interlacing coded blocks of qubits. A (noise-free) transversal gate is a unitary operator, whose action is nontrivial upon at most a single qubit in each coded block. For example, as portrayed in Fig. 1.3, we may encode the control qubit and the target qubit of a CNOT

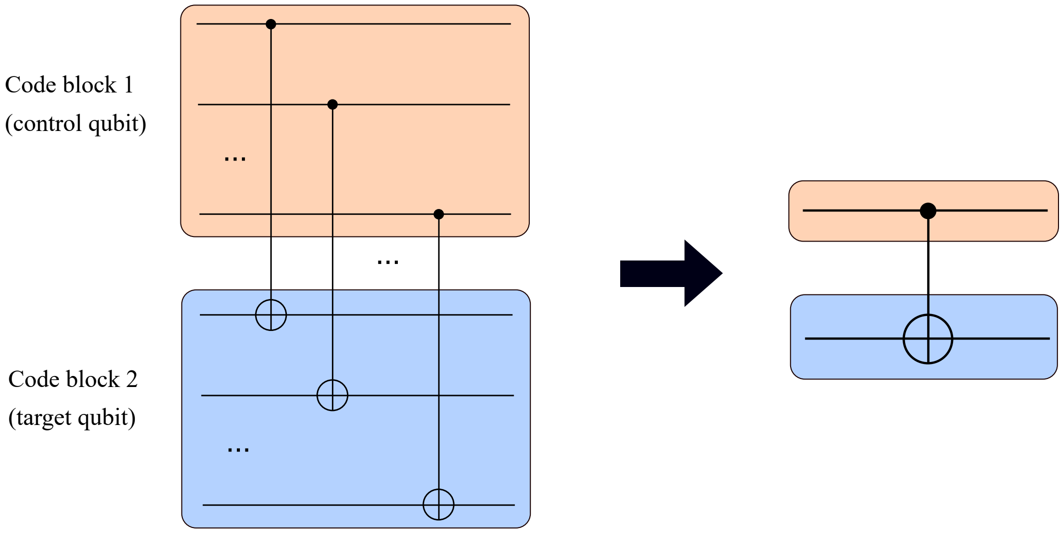


FIGURE 1.3: Illustration of the transversal implementation of a CNOT gate relying on two blocks of QECCs.

gate into two distinct set of physical qubits[2], which ensures that any single pre-existing error would only affect a single qubit in each code block at the output of the CNOT gate. To elaborate a little further, note that the code block 1 in (1.3) encodes all the control qubits, while the code block 2 encodes all the target qubits. Therefore, a single pre-existing error entering the CNOT gates would result in two errors at the output of the CNOT gates, but they fall into different code blocks, hence are correctable by the QECCs. However, whether a certain coded quantum gate can be implemented in the transversal manner depends on the specific code construction. The celebrated Eastin-Knill theorem [46] shows that no QECC can support a universal set of coded quantum gates. This motivates the search for specific QECCs that support transversal implementations of the largest set of quantum gates.

When error proliferation can be prevented, fault-tolerant quantum computation becomes possible, given that a sufficiently strong QECC is applied. This idea has been proved and termed as the **quantum threshold theorem** [23], formally stated as follows:

**Theorem 1.1** (Quantum Threshold Theorem)**.** *Using a QECC having the minimum distance of $d$, the logical error probability $p_{\mathrm{L}}$ can be upper bounded as follows [47]:*

$$p_{\mathrm{L}} \leq d^{\beta} \exp(-\alpha d), \tag{1.2}$$

*where $\alpha$ and $\beta$ are constants determined both by the specific code construction, as well as by the physical error probability[3], and the decoding algorithm. Specifically, if the physical error probability is lower than a **fault-tolerance threshold** $p_{\mathrm{T}}$, the parameter $\alpha$ is positive.*

---

[2] When a block of $n$ qubits are encoded into a code space of dimensionality $2^m$, we say that the code space is constituted by $m$ *logical qubits*, while the original $n$ qubits are referred to as *physical qubits*.

[3] The physical error probability of a coded quantum gate refers to the maximum probability that a certain component fails (the maximum failure probability among all components), for example, a single

Intuitively, the quantum threshold theorem indicates that once error proliferation is prevented, the error probability may be reduced exponentially as the code's minimum distance $d$ increases, provided that the physical error rate is lower than the fault-tolerance threshold. By contrast, the additional number of errors induced by the additional circuitry implementing QECCs having larger $d$ increases at most polynomially. Thus fault-tolerance is achievable, provided that a QECC having a sufficiently large $d$ is applied. Furthermore, the fault-tolerance threshold $p_\mathrm{T}$ should also be sufficiently high for the code to be practical.

In practice, the polynomially increasing factor $d^\beta$ of (1.2) might still be relatively high, forcing us to choose an extremely large $d$ to achieve the desired level of error reduction, which may be intractable in terms of computational resources. To reduce $d^\beta$, a common practice is to reduce the weight of the error-checking operations, which determines the number of additional gates required by the QECC [47]. Since these additional gates may themselves be perturbed by decoherence, a lower weight of check operators implies a smaller $d^\beta$.

Let us now briefly summarize the main criteria of "good QECCs" in terms of fault-tolerance:

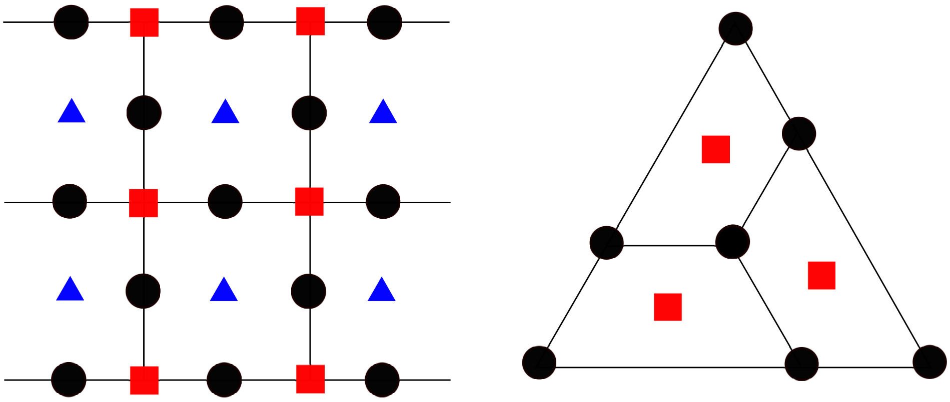
1. Support a large set of quantum gates to be implemented transversally to suppress error proliferation;
2. Have a high fault-tolerance threshold $p_\mathrm{T}$;
3. Have a small weight of error-checking operations, ensuring that $d^\beta$ is small;
4. The minimum distance $d$ increases fast with the code length $N$.

Finding a QECC satisfying all these criteria simultaneously is no doubt challenging. Fortunately, there has been a class of practical QECCs satisfying all expect the last criterion, namely the family of **quantum topological codes** [49,50]. These codes typically have a lattice-like structure, as demonstrated in Fig. 1.4. Specifically, in surface codes, the triangles in Fig. 1.4a represent the so-called $\mathcal{Z}$-stabilizers detecting the bit-flip errors within their neighbouring qubits (represented by circles), while the squares in Fig. 1.4a represent the $\mathcal{X}$-stabilizers detecting the phase-flip errors within their neighbouring qubits [51]. In colour codes, the squares in Fig. 1.4b represent both $\mathcal{Z}$-stabilizers and $\mathcal{X}$-stabilizers, since colour codes are so-called "self-dual" codes [51]. Observe that each error-checking operation[4] only involves a small number of qubits, corresponding to

---

gate fails, which is a component of a transversally implemented logical gate consisting of $n$ such gates [48, Sec. 10.6.1].

[4] The error-checking operations in Fig. 1.4 are also known as stabilizer checks [52]. Correspondingly, these codes are termed as quantum stabilizer codes [24, 52], which will be discussed in more detail in Section 3.2. There exist other constructions of quantum topological codes, including subsystem codes [53,54] and Floquet codes [55,56], whose error-checking operations may have even smaller weights.



(a) Surface codes

(b) Colour codes

FIGURE 1.4: Illustration of quantum topological codes. Circles represent physical qubits, while rectangles and triangles represent the check operations associated with their adjacent physical qubits.

small weights. Due to the small weights of error-checking operations, and an efficient decoding algorithm known as minimum-weight perfect matching [57], the fault-tolerance thresholds of quantum topological codes are typically as high as around 1% under practical error models [58]. Furthermore, the surface codes [49] shown on the left hand side support the transversal implementation of Pauli gates and the CNOT gate, while the colour codes shown on the right hand side support the transversal implementation of all Clifford gates [59].

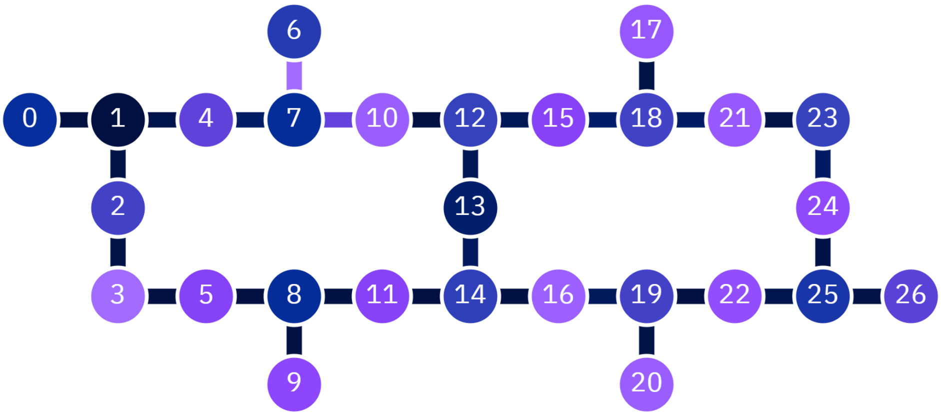


FIGURE 1.5: The qubit layout of IBM's 27-qubit superconducting quantum computer "IBM_Montreal" [60]. The circles represent the qubits, while the line segments between them represent the possible qubit interactions.

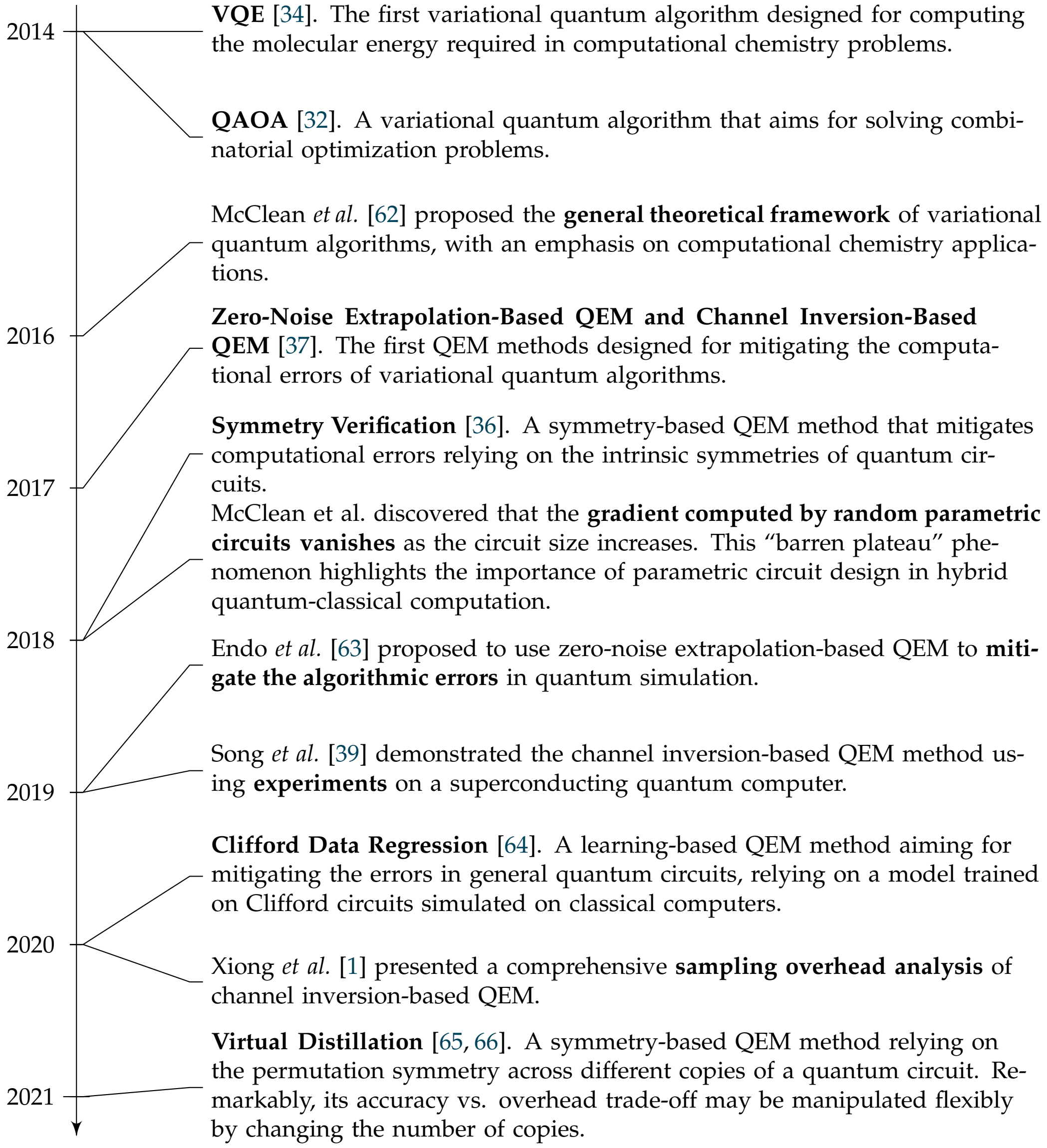
In addition to the aforementioned merits, the lattice-like structure of quantum topological codes also makes them particularly suitable for physical realizations. To elaborate, the state-of-the-art architecture of quantum computers can only support the so-called "nearest neighbour" interactions between qubits [61]. Consequently, any multi-qubit gate involving spatially distant qubits has to be compiled into a sophisticated set of nearest-neighbour gates. Hence they become impractical for the implementation of

QECCs, due to the extra errors induced by the additional circuitry. By contrast, the lattice-like structure of quantum topological codes fit nicely into the nearest neighbour architectures. For example, as portrayed in Fig. 1.5, one of IBM's most recent quantum computers, "IBM_Montreal", has a clear lattice-like qubit layout [60]. The numbered circles in Fig. 1.5 represent the qubits, while the line segments between them represent the possible qubit interactions that may be exploited to build two-qubit gates. Observe that the qubit layout in Fig. 1.5 is similar to that of the surface codes shown in Fig. 1.4a, although the lattice in Fig. 1.5 is still not sufficiently large to support a full block of the surface code. Nevertheless, since the number of qubits in quantum computers grows fast, we may expect that larger lattices that are capable of implementing quantum topological codes would be available in the near future.

### 1.2.2 Hybrid Quantum-Classical Computation and Quantum Error Mitigation

Although quantum topological codes have various beneficial characteristics as discussed in Section 1.2.1, they still may not be applicable to state-of-the-art noisy intermediate-scale quantum (NISQ) computers, which have limited numbers of qubits and physical error rates being potentially lower than the typical fault-tolerance thresholds of quantum topological codes [31]. Consequently, fault-tolerant quantum computation is not readily achievable at the time of writing.

In 2014, Peruzzo *et al.* [34] proposed a technique of sharing computational tasks between a classical computer and a quantum processor having a small number of qubits. This is later termed as the "variational quantum eigensolver (VQE)", which inspired a flourishing area of research, namely hybrid quantum-classical computation. As elegantly summarized in a later contribution by McClean *et al.* [62], the fundamental philosophy of hybrid quantum-classical computation is to formulate the computational task in a form that can be solved through **iterations between the classical computer and the quantum processor**. Specifically, the quantum processor typically contains a parametric circuit, which aims for evaluating a certain function, while the classical computer is responsible of determining the values of the parameters controlling the quantum circuit. By decomposing a large-scale computational task into iterations between smaller tasks, the function evaluation subroutine may fit nicely into NISQ devices. In particular, this small-scale subroutine might then be completed within the coherence time of the quantum processor, hence circumvents the requirement of full fault-tolerance. A timeline portraying the major milestones of hybrid quantum-classical computation and the associated error-resilience improvement techniques is shown in Fig. 1.6, and detailed as follows:



**VQE** [34]. The first variational quantum algorithm designed for computing the molecular energy required in computational chemistry problems.

**QAOA** [32]. A variational quantum algorithm that aims for solving combinatorial optimization problems.

McClean *et al.* [62] proposed the **general theoretical framework** of variational quantum algorithms, with an emphasis on computational chemistry applications.

**Zero-Noise Extrapolation-Based QEM and Channel Inversion-Based QEM** [37]. The first QEM methods designed for mitigating the computational errors of variational quantum algorithms.

**Symmetry Verification** [36]. A symmetry-based QEM method that mitigates computational errors relying on the intrinsic symmetries of quantum circuits.

McClean et al. discovered that the **gradient computed by random parametric circuits vanishes** as the circuit size increases. This "barren plateau" phenomenon highlights the importance of parametric circuit design in hybrid quantum-classical computation.

Endo *et al.* [63] proposed to use zero-noise extrapolation-based QEM to **mitigate the algorithmic errors** in quantum simulation.

Song *et al.* [39] demonstrated the channel inversion-based QEM method using **experiments** on a superconducting quantum computer.

**Clifford Data Regression** [64]. A learning-based QEM method aiming for mitigating the errors in general quantum circuits, relying on a model trained on Clifford circuits simulated on classical computers.

Xiong *et al.* [1] presented a comprehensive **sampling overhead analysis** of channel inversion-based QEM.

**Virtual Distillation** [65, 66]. A symmetry-based QEM method relying on the permutation symmetry across different copies of a quantum circuit. Remarkably, its accuracy vs. overhead trade-off may be manipulated flexibly by changing the number of copies.

FIGURE 1.6: Timeline of major milestones of hybrid quantum-classical computation, as well as the associated error-resilience improvement techniques. Bold fonts represent the name of the technique or the essential contribution of the research work.

- **VQE.** Early investigations concerning hybrid quantum-classical computation focus on the applications in the area of computational chemistry [67], partially due to the fact that computational chemistry problems are typically formulated as variational optimization problems, which admit a natural iterative form. More specifically, such problems often assume the following generic form

$$\min_{\boldsymbol{\theta}} \ \langle\psi(\boldsymbol{\theta})|\,\mathcal{H}\,|\psi(\boldsymbol{\theta})\rangle\,, \tag{1.3}$$

where $|\psi(\boldsymbol{\theta})\rangle$ denotes the quantum state produced by the parametric quantum circuit, $\mathcal{H}$ encodes the optimization problem, and the parameter $\boldsymbol{\theta}$ is determined by the classical computer, possibly using gradient-based methods [68] or gradient-free methods [69]. Due to the deep-rooted connection between hybrid quantum-classical algorithms and variational optimization, these algorithms are also termed as "variational quantum algorithms (VQAs)".

- **QAOA.** Interestingly, the popular quantum approximate optimization algorithm (quantum approximate optimization algorithm (QAOA)) of [32] was also proposed at a similar time as Peruzzo's seminal contribution, which was later recognized as a VQA, since it also obeys the form of (1.3). Furthermore, since QAOA aims for solving a wider range of problems than those of computational chemistry, namely combinatorial optimization problems, it soon received enormous attention from researchers in diverse areas [33, 70–73]. Remarkably, it has been applied to wireless communication-related problems as well, such as channel decoding [71].

- **QEM.** Even though VQAs require relatively short execution time, error control techniques are still indispensable for high-accuracy computation. In 2017, Temme *et al.* [37] proposed a pair of techniques for mitigating the error of VQAs, termed as quantum error mitigation (QEM), which are referred to at the time of writing as "**zero-noise extrapolation**" and "**channel inversion**" (or "probabilistic error cancellation"), respectively [74]. QEM is particularly suitable for VQAs executed on NISQ computers, since they typically require much less qubit overhead (sometimes no qubit overhead [37]) than QECCs. In particular, the error reduction performance of QEM has been demonstrated by Song et al. [39] using **experiments** on superconducting quantum devices.

  Due to its practical importance, QEM soon became an active area of research, encompassing four distinct types of techniques. Specifically, in addition to the zero-noise extrapolation and channel inversion techniques proposed in Temme's original contribution [37], **symmetry-based QEM** and **learning-based QEM** have been proposed in [36] and [64], respectively. Naturally, all these techniques have their own advantages and drawbacks, and may also be applied jointly for enhancing the overall error mitigation performance [65]. We will present a detailed introduction to these techniques in Chapter 3.

- **QEM for algorithm-level errors.** A somewhat surprising fact is that, apart from decoherence-induced errors, QEM may also be applied to mitigate algorithmic errors. For example, Endo *et al.* [63] proposed to use zero-noise extrapolation-based QEM to mitigate the computational errors inflicted by the Trotter approximation [75] in quantum simulation. This makes QEM a unique error-resilience improvement technique capable of mitigating both circuit-level and algorithm-level errors. By contrast, it is still unclear whether QECCs may be applied to correct algorithm-level errors.

- **The sampling overhead of QEM.** However, the low qubit-overhead of QEM methods does not come without a price. In fact, they have a different type of computational overhead, namely the sampling overhead [1,37,38]. Specifically, QEM methods typically rely on repeated activations of the quantum circuit, whose outcomes (termed as "samples" [38]) are then exploited for error mitigation in the post-processing stage. By contrast, in fault-tolerant computation facilitated by QECCs, the errors may be corrected within a single circuit execution, hence would not incur any sampling overhead, since we can locate the errors explicitly using the check operations.

  Mathematically speaking, the sampling overhead is related to the variance-escalation phenomenon caused by the non-unitary operations of QEM methods [38]. Briefly, as the depth of a quantum circuit increases, the variance of the computational results increases exponentially, hence the sampling overhead required for achieving a desired accuracy also grows dramatically. Ultimately, the benefits of QEM would be eroded by its sampling overhead, since the overall computational time becomes unacceptably long. Therefore, it is of significant practical value to explore how the sampling overhead required is influenced by various factors, such as the channel quality as well as the circuit size, and then develop techniques that strike a beneficial accuracy vs. sampling overhead trade-off for specific applications.

## 1.3 Novel Contributions

Against the aforementioned background, in this thesis, we aim for improving the performance of both circuit-level and algorithm-level QEM, in terms of their accuracy vs. sampling overhead trade-off. To this end, we commence by the analysis of the sampling overhead of channel inversion based QEM (CI-QEM). In this context, our main contributions [1] are as follows:

- We introduce the design philosophy of channel-inversion quantum error mitigation (CI-QEM), emphasizing its applications in hybrid quantum-classical computation (also known as variational quantum algorithms). Moreover, we quantify its sampling overhead, while providing insights into the variance-boosting phenomenon of CI-QEM. Specifically, we introduce the notion of sampling overhead factor (SOF) as a characteristic of quantum channels and the "basis operations" used in CI-QEM.

- We prove that **Pauli channels** have the **lowest SOF** among all non-coherent channels, and show that **depolarizing channels** have the lowest SOF among Pauli channels. This is in stark contrast to QECCs, for which the main computational overhead — the qubit overhead — achieves its maximum for depolarizing channels. Furthermore, we provide upper and lower bounds on the SOF of Pauli channels, showing that the SOFs of all Pauli channels are almost the same, when their average fidelity is close to 1.

- We additionally conceive a sophisticated amalgam of CI-QEM and quantum codes, including QECCs and quantum error detection codes (QEDCs), aiming for reducing the overall SOF. For QECCs, we show that there exist critical quantum circuit sizes, beyond which their amalgamation with CI-QEM is preferable to using pure CI-QEM. For QEDCs, we show that when the logical quantum gates are implemented transversally across physical qubits, using QEDCs alone is more favorable than using QEDC-CI-QEM, in terms of SOF. We also show that the QEDCs-CI-QEM scheme may still be useful for certain non-transversal quantum gate implementations.

Regarding the practical implementation of CI-QEM, we consider a Monte Carlo-based method. Compared to the idealistic exact channel inversion, the Monte Carlo-based channel inversion has a dramatically reduced pre-processing overhead, albeit this is achieved at the cost of a faster computational error escalation with respect to the circuit size. In this context, our main contributions [3] are:

- The Monte Carlo-based channel inversion method results in a less beneficial error scaling behaviour, but might exhibit a beneficial accuracy vs. overhead trade-off, since it reduces the pre-processing overhead. In order to better understand this trade-off, we propose bounds for the computational error scaling both in the absence of QEM and in the presence of Monte Carlo-based CI-QEM. Specifically, we show that in the absence of QEM, the error magnitude scales **linearly** with the number of gates $N_{\rm G}$, as well as with the gate error probability $\epsilon$, when we have $\epsilon N_{\rm G} \ll 1$. By contrast, we show that in the presence of Monte Carlo-based CI-QEM, the root mean squared error (RMSE) is upper bounded by the **square root** of $N_{\rm G}$ as well as $\epsilon$, when $\epsilon N_{\rm G} \ll 1$. This implies that when we use the same number of samples as the ideal CI-QEM (based on exact channel inversion), Monte Carlo-based CI-QEM can still provide a **quadratic error reduction** upon increasing $N_{\rm G}$, compared to the case of no QEM.
- We provide an intuitive geometric interpretation of the proposed error scaling laws by visualizing the decoherence-induced impairments on the Bloch sphere as the quantum circuit continues its operation.
- We illustrate the analytical results using various numerical examples. Specifically, we consider a practical application, namely multi-user detection in wireless communication systems using the quantum approximate optimization algorithm (QAOA), and show that our analytical results do apply.

Apart from CI-QEM, symmetry-based QEM is another class of circuit-level error mitigation techniques, which bears some resemblance in spirit to QEDCs. Recently, a new class of symmetry-based QEM methods, namely virtual distillation (virtual distillation (VD)) [65], has been proposed, which exploits the permutation symmetry of quantum states. Inspired by the philosophy of filter design techniques in classical signal processing theory, we propose the concept of "permutation filters", which incorporates the original VD as a special case. Specifically, our main contributions [2] are:

- We show that permutation filters would in general strike a more beneficial accuracy vs. sampling overhead trade-off than that of the original VD. We detail the general permutation filter design framework, including the functional form of the filters and the performance metric to be optimized.
- We propose an algorithm for optimal permutation filter design. In particular, we show that the local optimum of the optimization problem is unique, hence the globally optimal solution is attainable by the proposed algorithm. Furthermore, we show that permutation filters are particularly efficient in combating **narrowband** noise. Specifically, they are capable of providing an error-reduction improvement over permutation-based QEM methods, which scales polynomially with the noise bandwidth. We also show that the noise bandwidth decreases **exponentially** with the depth of the quantum circuit. This suggests that the proposed permutation filters can be used for supporting the employment of quantum circuits having an increased depth without degrading their fidelity.
- We quantify the error mitigation performance of permutation filters using a practical example, namely that of solving the multi-user detection problem in wireless communication systems relying on the quantum approximate optimization algorithm (QAOA). In particular, we show that the performance of permutation filters does not depend on the expected number of errors, which is a substantial benefit over the original VD.

From a broader perspective, the VD method may be viewed as exploiting the spatial consistency among different circuit activations. A natural question that arises is whether we could generalize the idea to the time domain, provided that some temporal consistency of the circuit may be verified. This motivates us to generalize the conventional state-oriented symmetry verification to **circuit-oriented symmetry verification**. In this context, our main contributions [5] are:

- A related topic, namely the superposition of causal orders [76–78], which can be physically realized using the quantum switch of [79], has been investigated from the perspective of quantum communication. For quantum circuits consisting of mutually commuting gates, we propose to use the original form of the quantum switch to verify the gate commutativity.

- For quantum circuits commuting with known operators, especially Pauli operators, we propose a modified quantum switch based method termed as the "**spatio-temporal stabilizer (spatio-temporal stabilizer (STS))**", which may be used for detecting and mitigating errors violating the commutativity condition. In contrast to conventional stabilizer-based symmetry verification, STSs do not depend on the specific quantum state, hence they are more generally applicable.
- We discuss the practical issues of implementing the STS method, including the simultaneous observability of STSs and their accuracy vs. overhead trade-off. We also provide quantum circuit designs that strike flexible accuracy vs. overhead trade-offs.
- We demonstrate the benefits of the STS method by applying it to practical quantum algorithms, including the quantum Fourier transform (quantum Fourier transform (QFT)) and the QAOA, where the conventional stabilizer checks are not applicable.

As for algorithm-level quantum error mitigation, we focus our attention on quantum phase estimation, which is a crucial subroutine in various practical quantum algorithms. Specifically, quantum phase estimation suffers from the issue of spectral leakage, which ultimately limits its computational accuracy. Aiming for mitigating the spectral leakage, our main contributions [4] are:

- We derive the Cramér-Rao bound (Cramér-Rao bound (CRB)) of the quantum phase estimation problem, which sheds light upon the asymptotic multiple-sample performance of practical estimators.
- We propose a dual-frequency estimator that asymptotically attains the CRB upon increasing the number of samples. We term this as the asymptotic regime.
- Using numerical simulations, we demonstrate that when the number of samples is sufficiently large, the proposed dual-frequency estimator outperforms the cosine window-based solution, which is known to exhibit the optimal accuracy in single-sample estimation.

## 1.4 Thesis Organization

The rest of this thesis is summarized in Fig. 1.7, and organized as follows:

- In **Chapter 2**, we present some preliminary knowledge about quantum information and quantum computation, which will be used extensively throughout the

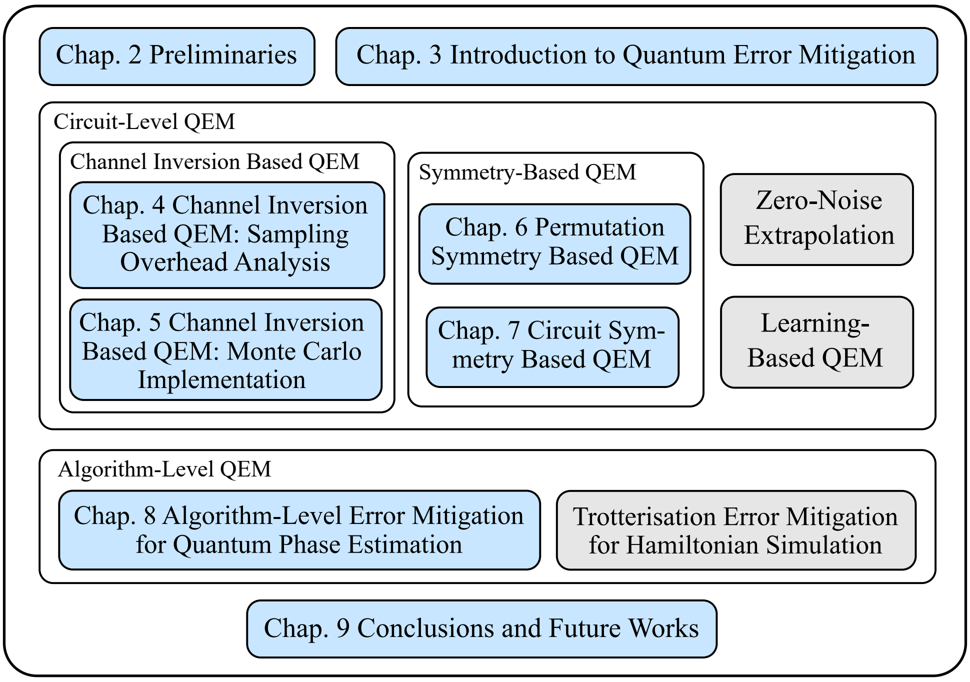


FIGURE 1.7: The structure of the thesis. Blue boxes represent the chapters of the thesis, while grey boxes represent related existing techniques in the literature.

thesis. We commence with a philosophical discussion on the fundamental units of both classical and quantum information in Section 2.1, with an emphasis on the view that quantum states are **physical realities**, instead of merely abstract information carriers. Then, in Section 2.2, we introduce the mathematical descriptions of pure states and the associated unitary operations from a geometrical perspective. This is followed by our discussions on mixed states in Section 2.3 and the corresponding general quantum operations in Section 2.4. Finally, in Section 2.5, we introduce a number of fundamental quantum algorithms that provide speedup over their classical counterparts, as well as hybrid quantum-classical algorithms tailored for noisy intermediate-scale quantum computers.

- In **Chapter 3**, we present an overview of quantum error mitigation. We commence with a discussion on the common error sources in quantum computation in Section 3.1. Then, in Section 3.2, we introduce quantum error correction codes and quantum error detection codes, which are the most extensively studied error-control techniques in the literature in addition to QEM. For circuit-level error sources, we introduce circuit-level QEM methods in Section 3.3, including channel inversion based QEM in Section 3.3.1, symmetry-based QEM in Section 3.3.2, zero-noise extrapolation based QEM in Section 3.3.3, and learning-based QEM in Section 3.3.4. For algorithm-level error sources, we introduce the corresponding algorithmic QEM methods in Section 3.4.
- In **Chapter 4**, we present a comprehensive analysis of the computational resources required by channel inversion based QEM, by providing rigorous upper and

lower bounds on its sampling overhead, concerning both coded and uncoded systems. Specifically, we introduce the quality metrics of quantum channels in Section 4.1, which are applied throughout this chapter. Then, we analyze the sampling overhead of uncoded and coded quantum gates in Section 4.2 and Section 4.3, respectively. This is followed by our numerical results quantifying the sampling overhead as a function of channel quality and other system parameters in Section 4.4.

- In **Chapter 5**, we consider a practical implementation of channel inversion based QEM relying on Monte Carlo sampling, and analyze its accuracy vs. sampling overhead trade-off. We first outline the system model in Section 5.1, followed by a comparison between the idealized exact channel inversion and Monte Carlo-based inversion in Section 5.2. Then, we analyze the error scaling behaviour of the Monte Carlo based CI-QEM in Section 5.3, and discuss the intuitions underlying these analytical results in Section 5.4. We continue by quantifying the accuracy vs. sampling overhead trade-off of both exact channel inversion and Monte Carlo based inversion in Section 5.5, using numerical examples including low-complexity single-qubit circuits and practical circuits for QAOA-aided multi-user detection.

- In **Chapter 6**, we conceive the permutation filtering method as a generalized version of virtual distillation (VD), which is a state-of-the-art symmetry-based QEM method. Specifically, we commence with a mathematical formulation of permutation symmetries and VD in Section 6.1. We then propose the general design framework of permutation filters in Section 6.2, including the functional form of permutation filters and a globally optimal design algorithm, followed by the analysis of their error reduction capability in Section 6.3. We quantify the performance of permutation filters using concrete examples in Section 6.4, where once again, we consider the QAOA-aided multi-user detection problem.

- In **Chapter 7**, we conceive circuit-oriented symmetry-based QEM methods. We first clarify the difference between state symmetries and circuit symmetries in Section 7.1, emphasizing that circuit symmetries may be more widely applicable than state symmetries. For the verification of gate commutativity, we propose to use quantum switches in Section 7.2. For circuits commuting with known unitaries, we propose the spatio-temporal stabilizer (STS) method in Section 7.3, including its general formulation in Section 7.3.1, its circuit implementation in Section 7.3.2, its unique simultaneous observability property in Section 7.3.3, and the accuracy vs. overhead trade-off in Section 7.3.4. We then continue by applying the STS method to QFT and QAOA circuits in Section 7.4, with a detailed discussion about the specific implementation strategies. The error mitigation performance of both quantum switches and STSs is quantified in Section 7.5.

- In **Chapter 8**, we conceive an algorithm-level error mitigation method for quantum phase estimation relying on a dual-frequency estimator. We first introduce the quantum phase estimation algorithm and the associated spectral leakage problem in Section 8.1, followed by its rigorous mathematical formulation and existing solutions in Section 8.2. Then, we present the CRB analysis of the quantum phase estimation problem in Section 8.3, which sheds light on the asymptotic performance of multi-sample estimators. Based on this analysis, we propose the novel dual-frequency estimator concept in Section 8.4, and quantify its estimation accuracy vs. sampling overhead trade-off in Section 8.5.

- In **Chapter 9**, we summarize our results, accompanied by a list of future research topics distilled from the previous chapters.

# Chapter 2

# Preliminaries

## 2.1 Classical vs. Quantum Information: An Ontological Perspective

Why are there beings at all instead of nothing? — Martin Heidegger

The *bit* is the most fundamental unit of information. But what do we really mean when we say "a bit", especially when we consider its practical representations in the physical world? Depending on the context, it might actually refer to one of the following two interpretations:

1. A logical variable taking one of two possible values (as implied by the name of "**bi**nary digi**t**");
2. A unit of the amount of information, as proposed in Shannon's landmark treatise [80].

The former interpretation (referred to later as "interpretation 1") is the one typically used in information processing and computation. In such applications, there exists some physical entity that has two possible states, which serves as the physical representation of the bit. By contrast, the latter (referred to later as "interpretation 2") is a rather abstract concept due to its independence of any physical substance, which finds its application in information compression and communication tasks.

The difference between the two interpretations is related to the problem, whether an object constitutes a part of the physical reality, or it is merely a piece of (incomplete) knowledge about physical entities. Philosophically speaking, interpretation 1 is the "ontic interpretation", while interpretation 2 is the "epistemic interpretation" [81]. This idea has been formalized in [82], where a state is said to be ontic, if there exists an injective mapping from the space of physical reality to the space of states, and is epistemic otherwise. To elaborate further, let us consider a simple example. Assume that we have a pair of biased coins, each has the probability of $p$ of being a head, where $p$ satisfies $-p\log_2 p - (1-p)\log_2(1-p) = 1/2$. The coins, being two-level "systems", may be viewed as a pair of ontic bits. However, the amount of information contained in this pair of coins is only one bit, which is equal to that of a single coin having equal probabilities of being head or tail. Therefore, "being head or tail" is an ontic state of the coin system, while the amount of information is only epistemic, since the same amount of information may correspond to substantially different physical entities.

In quantum information theory concerning information systems governed by quantum mechanics, the quantum bit, or in short, the qubit, may be viewed as the counterpart of the classical bit [48, Sec. 1.2]. By definition, a qubit is the space of the quantum state of a two-level quantum system [48, Sec. 1.2]. However, due to the peculiarity of quantum mechanics, a qubit differs from a classical bit in the sense that it may reside in a "superposition" of the two possible states. Moreover, whenever we wish to get access to the information stored in the qubit, we have to conduct a measurement, whose outcome is random.

The randomness of quantum measurements has led to a long-standing debate, which dates back to Einstein's famous EPR paradox [83]: whether a quantum state is an ontic state of the physical reality, or it is merely an epistemic state? This problem is particularly relevant to quantum information, as quantum states would be unreliable carriers of information if they are epistemic, since a single physical entity may be recognized as different quantum states, hence incur ambiguities in information processing tasks. Recently, this problem has been partially resolved in [84], which shows that quantum states are indeed ontic under some mild assumptions. Notably, this result is also related to the celebrated Bell's theorem [85], which states that locality and realism cannot be satisfied simultaneously for theories that are consistent with quantum mechanics.

Now that we acknowledged that qubits may be viewed as a generalization of classical bits under the ontic interpretation, a natural question that arises is whether a generalization in the sense of "the amount of quantum information" is possible. It turns out that, such a generalization is not yet known at the time of writing, to the best of the author's knowledge. Consequently, the qubit cannot be viewed as a unit of information. Rather, it is the mathematical abstraction of the basic physical entity exploited in information processing tasks that rely on (or benefit from) the laws of quantum mechanics. For example, we may conceive efficient algorithms relying on qubits by exploiting their unique characteristics, but we cannot say "this message contains 100 qubits of information".

To conclude, qubits are essentially physical entities (or their mathematical abstractions) rather than entities of pure information. In the rest of this thesis, whenever we refer to qubits, we mean either the physical realizations of qubits, or their mathematical models, which will be clear from the context.

## 2.2 Pure States, Unitary Operations, and the Bloch Sphere

As discussed in Section 2.1, the basic carrier of information in quantum mechanical information processing systems is a qubit, namely a two-level quantum system. Ideally, the state of a qubit can be characterized by a vector in a two-dimensional Hilbert space as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{2.1}$$

where $\alpha$ and $\beta$ are complex numbers satisfying the normalization property of $|\alpha|^2 + |\beta|^2 = 1$. Quantum states taking the form of (2.1) are also referred to as **pure states**. Under the conventional computational basis [48, Sec. 1.2], the basis vectors $|0\rangle$ and $|1\rangle$ can be expressed as

$$|0\rangle = [1,\ 0]^{\mathrm{T}},\ \ |1\rangle = [0,\ 1]^{\mathrm{T}}.$$

According to the Schrödinger equation representing the evolution of closed quantum systems [86], an ideal operation acting on a quantum state can be modeled as a unitary transformation. In the case of a qubit, the unitary transformation may be represented by a 2 × 2 unitary matrix. For example, the widely-used bit-flip gate (also known as the X-gate) satisfies the following conditions

$$\mathcal{X}\,|0\rangle = |1\rangle\,,\ \mathcal{X}\,|1\rangle = |0\rangle\,. \tag{2.2}$$

Therefore, under the computational basis, the X-gate may be represented in the following matrix form

$$\boldsymbol{S}_{\mathcal{X}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \tag{2.3}$$

which is a 2 × 2 unitary matrix.

In order to acquire the information stored in a quantum state, one may perform a measurement of the state. The measurement outcome depends on the basis on which the measurement is performed, and it is essentially probabilistic [86]. For example, when measured on the computational basis, the measurement outcome of a qubit state given by (2.1) can be either 0 or 1, with probability $|\alpha|^2$ and $|\beta|^2$, respectively. In light of this, we see that states taking the following form

$$|\psi(\theta)\rangle = e^{\jmath\theta}(\alpha\,|0\rangle + \beta\,|1\rangle), \tag{2.4}$$

having identical $\alpha$ and $\beta$ (thus differ only by $\theta$), cannot be distinguished from one another using measurement. In fact, the only way of characterizing a quantum state is

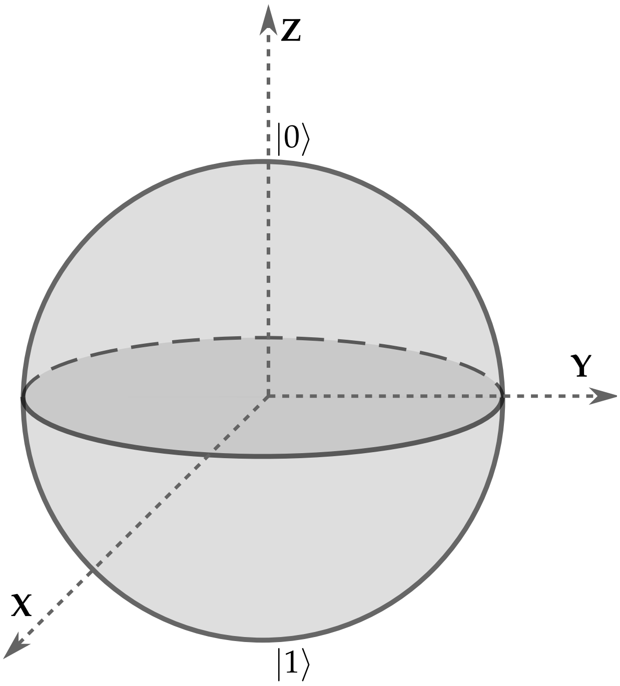


FIGURE 2.1: The Bloch sphere representing the space of all single-qubit pure states.

through measurement, and hence the states in (2.4) are regarded as mutually equivalent. Following a similar line of reasoning, the global phase in an ideal (unitary) quantum operation is also irrelevant, implying that

$$\tilde{\boldsymbol{G}}(\theta) = e^{\jmath\theta}\boldsymbol{G},\ \boldsymbol{G}\boldsymbol{G}^{\mathrm{H}} = \boldsymbol{G}^{\mathrm{H}}\boldsymbol{G} = \boldsymbol{I}, \tag{2.5}$$

are regarded as mutually equivalent. Mathematically, this means that the ensemble of all ideal single-qubit quantum operations, denoted by $G(1)$, is characterized by

$$G(1) = U(2)/U(1) = SU(2), \tag{2.6}$$

where $U(n)$ denotes the unitary group of dimension $n \times n$, $SU(n)$ denotes the special unitary group of dimension $n \times n$, and $C = A/B$ means that group $C$ is the quotient group of groups $A$ and $B$ [87]. Then equation (2.6) leads to the celebrated geometric interpretation of single-qubit quantum states — the Bloch sphere, as shown in Fig. 2.1. To elaborate, the ideal single-qubit quantum operations can be described using the special orthogonal group $SO(3)$, which corresponds to three-dimensional rotations, due to the $SU(2)$–$SO(3)$ homomorphism [87]. Since the three-dimensional vectors that the rotations act upon constitute the unit sphere, it can also be shown that pure quantum states lie on the unit sphere, again termed as the Bloch sphere in the literature [48, Sec. 1.2]. Using the Bloch sphere, the widely-used X-, Y- and Z-gates given by

$$\boldsymbol{S}_{\mathcal{X}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ \boldsymbol{S}_{\mathcal{Y}} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix},\ \boldsymbol{S}_{\mathcal{Z}} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{2.7}$$

can now be interpreted as $\pi$ rotations around the $X$, $Y$ and $Z$ axes of the Bloch sphere, respectively. Furthermore, the basis states $|0\rangle$ and $|1\rangle$ correspond to the North Pole and the South Pole of the Bloch sphere, respectively. The general single-qubit states can be obtained by rotating the vectors corresponding to the basis states, hence correspond to the points on the surface of the Bloch sphere.

In general, a pure state of an $n$-qubit system resides in a $2^n$-dimensional vector space. For example, under the computational basis, a two-qubit state $|00\rangle$ can be written as follows

$$\begin{aligned} |00\rangle &= |0\rangle \otimes |0\rangle \\ &= [0,\ 0,\ 0,\ 1]^{\mathrm{T}}. \end{aligned} \tag{2.8}$$

All two-qubit pure states can thus be represented as the linear combinations of four basis states $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$. Similarly, any $n$-qubit state can be represented as the linear combinations of $2^n$ basis states, which corresponds to all $n$-bit binary strings.

## 2.3 Mixed States and Density Matrices

In practice, the physical systems representing the qubits are not closed systems. Consequently, the interaction between the qubits and the environment will cause decoherence, namely turning a deterministic pure quantum state into a probabilistic mixture of states. The probabilistic mixtures are also termed as **mixed states**. A mixed state of a qubit can be fully characterized by a $2 \times 2$ matrix termed as the density matrix $\rho$ given by

$$\rho = \sum_i p_i \left|\psi_i\right\rangle \left\langle\psi_i\right|, \tag{2.9}$$

satisfying $p_i \geq 0$ for all $i$ and $\sum_i p_i = 1$. Hence the density matrix is positive semi-definite and has unit trace. The pure states $|\psi_i\rangle$ are the components of the probabilistic mixture. Additionally, under the computational basis, it can be expressed as the linear combination of the following matrices:

$$\begin{aligned} S_{\mathcal{I}} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ S_{\mathcal{X}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \\ S_{\mathcal{Y}} &= \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix},\ S_{\mathcal{Z}} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \end{aligned}$$

The density matrix representation is also compatible with pure states. For example, in the density matrix form, the basis state $|0\rangle$ can be written as $|0\rangle \langle 0|$, which may be viewed as a special case of (2.9) having only a single component.

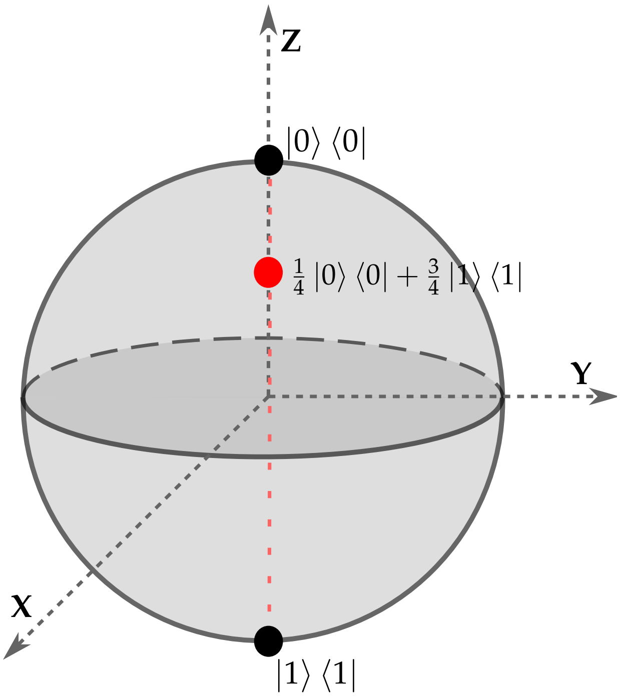


FIGURE 2.2: The interior of the Bloch sphere represents the space of all single-qubit states. The mixed states correspond to the convex combination of pure states residing on the surface of the sphere.

Geometrically, single-qubit mixed states can also be represented using the Bloch sphere. In particular, let us consider a simple example:

$$\rho_1 = \frac{1}{4}|0\rangle\langle 0| + \frac{3}{4}|1\rangle\langle 1|. \tag{2.10}$$

The state $\rho_1$ may be represented as the **vector summation** of the vectors corresponding to the pure states $|0\rangle\langle 0|$ and $|1\rangle\langle 1|$. This implies that $\rho_1$ corresponds to a point within the Bloch sphere, as illustrated in Fig. 2.2. In fact, all single-qubit mixed states correspond to points within the Bloch sphere, which may be observed from (2.9). To elaborate a little further, the pure states $|\psi_i\rangle\langle\psi_i|$ lie on the surface of the Bloch sphere, and the mixed state $\rho$ may be shown to be the convex combination of the pure states, since we have $p_i \geq 0$ for all $i$ and $\sum_i p_i = 1$. Therefore, the mixed states lie in the convex hull of pure states, which is exactly the interior of the Bloch sphere.

In general, a mixed state of an $n$-qubit system can be characterized by a $2^n \times 2^n$ density matrix. Similar to the single qubit case, the $n$-fold tensor products of $S_{\mathcal{I}}$, $S_{\mathcal{X}}$, $S_{\mathcal{Y}}$ and $S_{\mathcal{Z}}$ form a basis for the space of $2^n \times 2^n$ density matrices, termed as the $n$-qubit Pauli group. To facilitate further discussion, we denote $S_i^{(n)}$ as the $i$-th operator in the $n$-qubit Pauli group. The superscript $(n)$ is omitted when it is clear from the context.

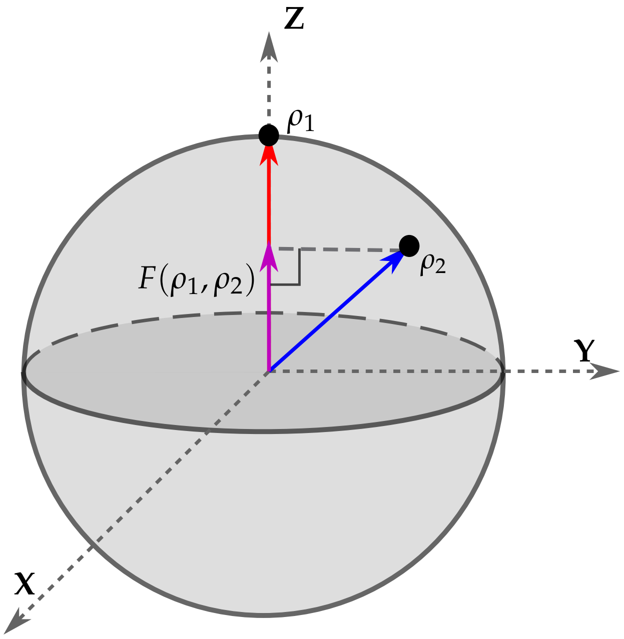


FIGURE 2.3: The fidelity between a pure state and a (potentially) mixed state can be represented by the inner product of the vectors representing the states in the Bloch sphere.

The difference between two quantum states $\rho_1$ and $\rho_2$ is typically quantified using the terminology of fidelity, defined as [48, Section 9.2.2]

$$F(\rho_1, \rho_2) = \mathrm{Tr}\left\{\left(\rho_1^{\frac{1}{2}}\rho_2\rho_1^{\frac{1}{2}}\right)^{\frac{1}{2}}\right\}^2, \tag{2.11}$$

and can be simplified as

$$F(\rho_1, \rho_2) = \mathrm{Tr}\{\rho_1^{\dagger}\rho_2\}$$

if either $\rho_1$ or $\rho_2$ represents a pure state. On the Bloch sphere, the fidelity between a pure state and a (potentially) mixed state is given by the inner product of the vectors representing the states, as illustrated in Fig. 2.3.

## 2.4 General Quantum Operations and Their Representations

The unitary operations of (2.5) (and exemplified by (2.7)) introduced in Section 2.2 map a pure state to another pure state, hence cannot be applied to model the decoherence process, which maps a pure state to a mixed state. In quantum computers, except for measurements, a general quantum operation (carried out by a "quantum gate") is modeled as a completely positive, trace-preserving (CPTP) transformation [48, Sec. 8.2].

A widely-used representation of a general quantum operation is termed as the operator-sum representation, also known as the Kraus representation [48, Sec. 8.2]. In particular, using the operator-sum representation, a quantum operation may be written as the following mapping

$$\mathcal{E}(\rho) = \sum_k \boldsymbol{E}_k \rho \boldsymbol{E}_k^\dagger, \tag{2.12}$$

where the matrices $\boldsymbol{E}_k$ are referred to as the operation elements of the quantum operation $\mathcal{E}(\rho)$, which are not required to be unitary. Intuitively, the operation $\mathcal{E}(\rho)$ maps its input $\rho$ to one of the potential outputs $\boldsymbol{E}_k \rho \boldsymbol{E}_k^\dagger$ with some probability, and the equivalent output is the probabilistic mixture $\sum_k \boldsymbol{E}_k \rho \boldsymbol{E}_k^\dagger$. In the classical case, probabilities are required to be normalized to one. Similarly, in the quantum case, the trace of the output should be normalized to one, namely we have

$$\operatorname{Tr}\left\{\sum_k \boldsymbol{E}_k \rho \boldsymbol{E}_k^\dagger\right\} = 1. \tag{2.13}$$

Since this relationship is satisfied for all input $\rho$, we further have

$$\sum_k \boldsymbol{E}_k \rho \boldsymbol{E}_k^\dagger = \boldsymbol{I}. \tag{2.14}$$

This is the so-called **completeness relation** [48, Sec. 8.2].

Without loss of generality, we may decompose a quantum operation into an ideal unitary operation and another CPTP operation, as follows

$$\begin{aligned}\mathcal{E}(\rho) &= \mathcal{C}(\mathcal{U}(\rho)) \\ &= \sum_k \boldsymbol{E}_k \boldsymbol{U} \rho \boldsymbol{U}^\dagger \boldsymbol{E}_k^\dagger,\end{aligned} \tag{2.15}$$

where the matrix $\boldsymbol{U}$ is the only operation component of the unitary operation $\mathcal{U}$, and the matrices $\boldsymbol{E}_k$ denote the operation components of the operation $\mathcal{C}$. In the context of quantum computation, the unitary operation $\mathcal{U}$ models the error-free operation that we intend to implement, while the CPTP operation $\mathcal{C}$ models the imperfection of our implementation, which is also popularly referred to as the **quantum channel**.

Alternatively, quantum operations can also be expressed in a matrix form. To elaborate, if we represent a quantum state having a density matrix of $\rho$ as a vector $\boldsymbol{x}$, a quantum operation $\mathcal{C}$ acting on $\boldsymbol{x}$ having the following operator-sum representation

$$\mathcal{C}(\rho) = \sum_i \boldsymbol{K}_i \rho \boldsymbol{K}_i^\dagger$$

can be written as a matrix $\boldsymbol{C}$ satisfying

$$\boldsymbol{C}\boldsymbol{x} = \sum_i (\boldsymbol{K}_i^* \otimes \boldsymbol{K}_i)\boldsymbol{x}, \tag{2.16}$$

implying that

$$\boldsymbol{C} = \sum_i (\boldsymbol{K}_i^* \otimes \boldsymbol{K}_i).$$

In general, the specific matrix form of a quantum channel depends on the set of bases we choose. In this thesis, we use the Pauli transfer matrix (PTM) representation of quantum operations, given by

$$[\boldsymbol{C}]_{i,j} = \frac{1}{2^n}\mathrm{Tr}\{\boldsymbol{S}_i\mathcal{C}(\boldsymbol{S}_j)\}. \tag{2.17}$$

In this sense, the vector representation $\boldsymbol{x}$ of a density matrix $\rho$ can be expressed as

$$[\boldsymbol{x}]_i = \frac{1}{\sqrt{2^n}}\mathrm{Tr}\{\boldsymbol{S}_i\rho\}. \tag{2.18}$$

We choose the PTM representation mainly because many important properties of a quantum operation may be readily obtained from its PTM. In general, the PTM representation of any completely positive trace-preserving (CPTP) quantum operation may be expressed as follows:

$$\boldsymbol{C} = \begin{bmatrix} 1 & \boldsymbol{0}^{\mathrm{T}} \\ \boldsymbol{b} & \tilde{\boldsymbol{C}} \end{bmatrix}. \tag{2.19}$$

To elaborate, the top-left entry is always 1, since $\frac{1}{2^n}\mathrm{Tr}\left\{\boldsymbol{I}\mathcal{C}(\boldsymbol{I})\right\} = 1$, which follows from the fact that $\mathcal{C}$ preserves trace. Similarly, the rest of the first row is constituted by all zeros, since all other Pauli operators (i.e. except $\boldsymbol{I}$) have zero trace.

Correspondingly, a quantum state (either pure or mixed) has the following PTM representation

$$\boldsymbol{x} = [1,\ \tilde{\boldsymbol{x}}]^{\mathrm{T}}, \tag{2.20}$$

which may be viewed as a point in the $(2^n - 1)$-dimensional space under the homogeneous coordinates [88].[1] We may further obtain

$$\boldsymbol{C}\boldsymbol{x} = [1,\ \tilde{\boldsymbol{C}}\tilde{\boldsymbol{x}} + \boldsymbol{b}]^{\mathrm{T}}, \tag{2.21}$$

which implies that the effect of a quantum channel $\boldsymbol{C}$ on the state $\boldsymbol{x}$ may be viewed as an affine transformation, consisting of the linear transformation $\tilde{\boldsymbol{C}}$ and the translation $\boldsymbol{b}$.

---

[1] The conventional $(2^n - 1)$-dimensional vector representation of a point cannot accommodate translation as a linear operation. Fortunately, by using the $2^n$-dimensional homogeneous coordinates (under which the first entry of the vector is always 1, while the remaining entries correspond to the conventional Euclidean coordinates), the translation becomes a linear operation [88].

## 2.5 Quantum Algorithms

In 1982, Richard Feynman proposed the idea of simulating quantum systems using quantum computers in his seminal speech, *Simulating Physics with Computers* [10]. Feynman believed that quantum systems could be simulated more efficiently by quantum algorithms than by classical algorithms. Since then, this belief has been developed into a fruitful and active field of research, supported by concrete theorems and experiments.

The potential of quantum algorithm in efficiently solving practical problems other than simulating physical systems was demonstrated by Peter Shor [12], who proposed the first quantum algorithm capable of providing *exponential* speedup compared to its best known classical counterpart. Soon after Shor's discovery, Lov Grover proposed a quantum algorithm providing a quadratic speedup for search problems, which is applicable even to NP-hard problems [13]. Later investigations also contributed to the library of quantum algorithms providing substantial speedup, including the Harrow-Hassidim-Lloyd (HHL) algorithm for solving linear systems [16], and quantum walk-based algorithms [89, 90].

In this section, we discuss the fundamental subroutines of these successful quantum algorithms. Furthermore, we will also introduce recent advances in the area of hybrid quantum-classical algorithms tailored for NISQ computers.

### 2.5.1 Fundamental Subroutines Providing Quantum Speedup

Broadly speaking, quantum algorithms providing substantial speedup over their classical counterparts fall into three categories.

The first relies on the capability of quantum algorithms for finding periodicity. Specifically, the quantum version of the classical discrete Fourier transform is exponentially faster than its classical counterpart [48, Section 5.1]. The quantum Fourier transform is typically used as a subroutine of quantum phase estimation [91], which aims for computing the eigenvalue associated with a given eigenvector of a matrix. To elaborate a little further, for an eigenstate $|\psi\rangle$ of a given unitary matrix $\boldsymbol{U}$, we have

$$\boldsymbol{U}\,|\psi\rangle = e^{j\varphi}\,|\psi\rangle\,, \tag{2.22}$$

where $e^{j\varphi}$ is the eigenvalue associated with $|\psi\rangle$, and $\varphi$ is the *phase* to be estimated. As portrayed in Fig. 2.4, the phase estimation process commences with preparing the uniform superposition $|+\rangle^{\otimes M}$ over all $2^M$ possible states in the ancillary register, using

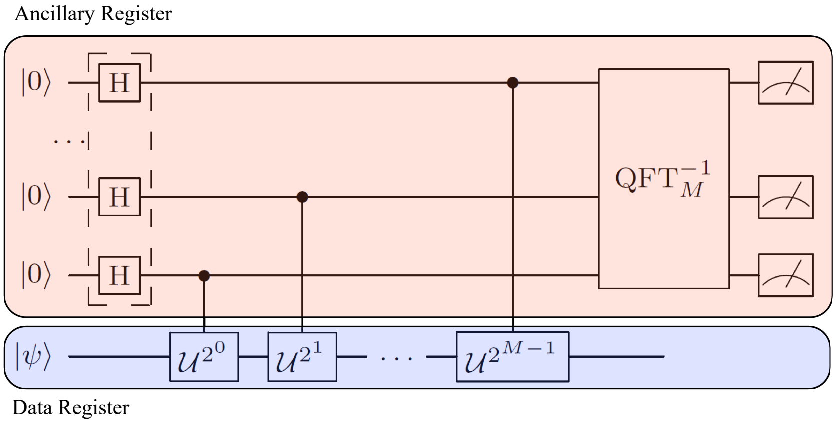


FIGURE 2.4: The circuit diagram of quantum phase estimation.

Hadamard gates given by[2]

$$\boldsymbol{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Then, by repeatedly applying the controlled-$\mathcal{U}$ operations in Fig. 2.4 for $N$ times, one may obtain the following sinusoidal data record

$$\boldsymbol{y} = [1,\ e^{j\varphi},\ e^{j2\varphi}, \ldots,\ e^{jN\varphi}]^{\mathrm{T}}. \tag{2.23}$$

Intuitively, the initial state $|+\rangle^{\otimes M}$ may be viewed as a rectangular window applied to the data record, which will be discussed in more detail in Chapter 8. Observe that the phase parameter $\varphi$ is related to the frequency of the record $\boldsymbol{y}$, which is $\varphi/2\pi$. Thus for $N = 2^M - 1$, we may store the accumulated phase of $\boldsymbol{y}$ in the $M$-qubit ancillary register in Fig. 2.4, and estimate $\varphi$ by applying the quantum Fourier transform to this register. This is essentially similar to the technique of estimating the frequency of a sinusoidal signal by computing its classical discrete Fourier transform and finding the peak of the spectrum, which is widely applied in classical signal processing [92].

This approach can be extended to an arbitrary matrix $\boldsymbol{A}$, by constructing an ancillary unitary matrix $\boldsymbol{U} = e^{j\boldsymbol{A}}$. Due to the superposition property of quantum states, when the input state is not an eigenvector, quantum phase estimation would return a superposition state of multiple eigenvalues. This is useful in some quantum algorithms designed for solving linear algebraic problems [16].

Typically, the computational complexity of quantum phase estimation is proportional to $N$. The phase estimation error is on the order of $O(1/N)$, which implies a quadratic speedup compared to classical phase estimation, since classically the estimation error scales as $O(1/\sqrt{N})$. This is related to a deep-rooted characteristic of quantum information systems, namely the Heisenberg limit [93]. However, the $N$-th power of $\boldsymbol{U}$ does

---

[2] The "plus state" $|+\rangle$ is defined as $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, which may be obtained by applying a Hadamard gate to the $|0\rangle$ state.

not necessarily require applying $\boldsymbol{U}$ $N$ times, hence the actual speedup may be well beyond quadratic, sometimes even exponential. A representative example of the super-quadratic speedup is Shor's algorithm for factoring integers [12]. In Shor's algorithm, the $N$-th power of the unitary operator $\boldsymbol{U}$ corresponds to the modular exponentiation of integers, which admits efficient implementations. Consequently, Shor's algorithm provides an exponential speedup over its best known classical counterpart.

The second category relies on the fact that quantum states can accumulate their probability amplitude rapidly, which may be viewed as generalizing the quadratic speedup provided by quantum phase estimation. A representative algorithm of this kind is the quantum amplitude amplification, first proposed by Grover for finding a marked entry in an unstructured database [13]. Geometrically speaking, quantum amplitude amplification iterates between a pair of reflection operations. To elaborate a little further, in each iteration of quantum amplitude amplification, the quantum state is first reflected about a hyperplane representing the "marked subspace", using the following operator:

$$\mathcal{U}_{\mathrm{o}} = \boldsymbol{I} - 2 \sum_{k \text{ marked}} |k\rangle \langle k| , \tag{2.24}$$

and then reflected with respect to the initial state $|\psi\rangle_{\mathrm{init}}$, using the operator

$$\mathcal{U}_{\mathrm{s}} = 2\,|\psi\rangle_{\mathrm{init}} \langle\psi|_{\mathrm{init}} - \boldsymbol{I}. \tag{2.25}$$

The operator $\mathcal{U}_{\mathrm{o}}$ is also known as the **oracle** [48, Section 6.1], which may be viewed alternatively as a phase flip on all marked entries as follows:

$$\mathcal{U}_{\mathrm{o}} |k\rangle = \begin{cases} -|k\rangle , & k \text{ is marked;} \\ |k\rangle , & \text{otherwise.} \end{cases} \tag{2.26}$$

We may understand the quadratic speedup of quantum amplitude amplification by considering its state evolution on a two-dimensional plane. Specifically, the initial state may be decomposed as follows

$$|\psi\rangle_{\mathrm{init}} = \sin\theta\,|\psi\rangle_{\mathrm{mark}} + \cos\theta\,|\psi\rangle_{\mathrm{rest}} , \tag{2.27}$$

where $|\psi\rangle_{\mathrm{mark}}$ and $|\psi\rangle_{\mathrm{rest}}$ denote the projections of the initial state onto the marked subspace and its orthogonal complement, respectively. After each iteration, the quantum state is rotated by an angle of $2\theta$ on the plane spanned by $|\psi\rangle_{\mathrm{mark}}$ and $|\psi\rangle_{\mathrm{rest}}$ [94], as portrayed in Fig. 2.5. Hence the output state after $n$ iterations is given by [48, Section 6.1]

$$|\psi\rangle_{\mathrm{out}}^{(n)} = \sin[(2n+1)\theta]\,|\psi\rangle_{\mathrm{mark}} + \cos[(2n+1)\theta]\,|\psi\rangle_{\mathrm{rest}} . \tag{2.28}$$

This implies that the success probability of the search algorithm will be close to 1 when the number of iterations is set to $\lfloor \pi/4\theta \rfloor$, which is on the order of $O(\sqrt{N/M})$, where

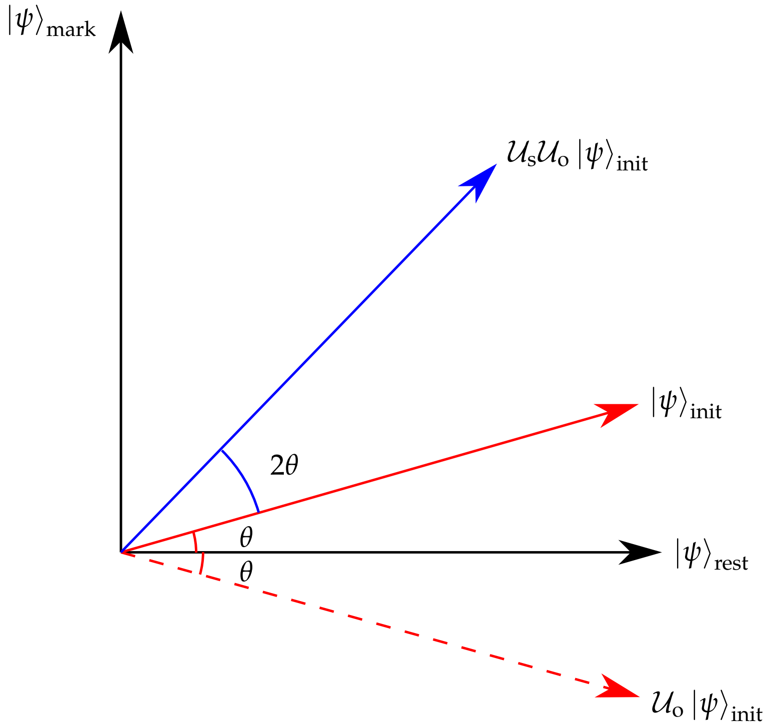


FIGURE 2.5: Geometric illustration of the quantum amplitude amplification.

$N$ is the number of all candidate solutions, and $M$ denotes the number of marked solutions (i.e. the dimensionality of the marked subspace).

Let us now consider the classical exhaustive search, whose expected complexity would be on the order of $O(N/M)$. We may also view this procedure as a state evolution over a two-dimensional plane. Specifically, the success probability of the "initial state" is $p_{\text{init}} = M/N$, which may be alternatively represented as

$$\boldsymbol{x}_{\text{init}} = p_{\text{init}} \cdot \boldsymbol{x}_{\text{mark}} + (1 - p_{\text{init}})\boldsymbol{x}_{\text{rest}}, \tag{2.29}$$

where $\boldsymbol{x}_{\text{mark}}$ and $\boldsymbol{x}_{\text{rest}}$ may be arbitrary orthonormal basis vectors of the two-dimensional plane. As an example, we choose $\boldsymbol{x}_{\text{mark}} = [1,\ 0]^{\text{T}}$ and $\boldsymbol{x}_{\text{rest}} = [0,\ 1]^{\text{T}}$. In each iteration, we randomly choose one entry and test whether it is a marked entry, hence the output state after $n$ iterations is given by

$$\boldsymbol{x}^{(n)} = (np_{\text{init}}) \cdot \boldsymbol{x}_{\text{mark}} + (1 - np_{\text{init}})\boldsymbol{x}_{\text{rest}}. \tag{2.30}$$

Upon comparing (2.30) to (2.28), we see that the associated quantum operations can accumulate the probability amplitude of quantum states linearly, whereas the classical search can only accumulate the probability linearly. Note that probability amplitudes are in fact the square roots of probabilities, hence quantum amplitude amplification is capable of providing a quadratic speedup.

The third category is what Richard Feynman originally conceived in his vision of quantum computation, namely simulating quantum physical systems, also known as Hamiltonian simulation [11]. Mathematically speaking, Hamiltonian simulation aims for approximating the matrix exponential $\exp(-jt\mathcal{H})$, where $\mathcal{H}$ is the Hamiltonian, which may be expressed in terms of a $2^n \times 2^n$ matrix. Classically, one would expect a computational complexity on the order of $O(2^n)$ for the same task. By contrast, when the Hamiltonian $\mathcal{H}$ is *sparse* under a certain representation, a polynomial-time quantum algorithm is typically viable. A possible such representation is constituted by a linear combination of sub-Hamiltonians as follows:

$$\mathcal{H} = \sum_{k=1}^{K} \theta_k \mathcal{H}_k, \tag{2.31}$$

where for each $\mathcal{H}_k$, the corresponding exponentiation $\exp\{j\tau\theta_k\mathcal{H}_k\}$ can be readily implemented by native gates in quantum computers. Although the sub-Hamiltonians are not in general mutually commutative, one may use the Trotter approximation [75] to achieve an accuracy of $O(1/\Delta t)$, where $\Delta t = t/N$ is the length of an evolution time step. Therefore, the accuracy can be arbitrarily improved as long as one opts for a sufficiently large number of time steps $N$. The total computational complexity is on the order of $O(KN/\epsilon)$, where $\epsilon$ is the maximum tolerable error.

Another common sparsity is that the Hamiltonian has only a small number of non-zero entries. If additionally we can compute each of the non-zero entries efficiently (i.e. using a constant number of operations per entry) using an oracle, quantum walk-based methods may be applied to achieve a computational complexity on the order of $O(td\|\mathcal{H}\|_{\max}/\sqrt{\epsilon})$ [95], where $d$ is the maximum number of non-zero entries per row/column, and $\|\mathcal{H}\|_{\max}$ represents the maximum absolute value of the entries in $\mathcal{H}$. Specifically, the quantum walk based methods rely on a walk operator $\mathcal{W}$ having the following eigendecomposition [95]

$$\mathcal{W} = j\mathcal{S}\left(2\mathcal{T}\mathcal{T}^\dagger - \mathcal{I}\right), \tag{2.32}$$

where $\mathcal{T}$ is an isometry that maps an $n$-qubit computational basis state $|i\rangle$ to two $n$-qubit quantum registers (plus one ancilla), given by [95]

$$\mathcal{T} = \sum_{i=1}^{2^n} |\eta_i\rangle \langle i|, \tag{2.33}$$

where we have

$$|\eta_i\rangle = \sqrt{\frac{\theta}{\|\mathcal{H}\|_1}} \sum_{k=1}^{2^n} \sqrt{\mathcal{H}_{ik}^*}\, |k\rangle + \sqrt{1 - \frac{\theta \sum_{k=1}^{2^n} |\mathcal{H}_{ik}|}{\|\mathcal{H}\|_1}}\, |2^n + 1\rangle ,$$

with $\theta$ being a parameter controlling the step size of state evolution. The operator

$\mathcal{S}$ simply swaps the two quantum registers, meaning that $\mathcal{S}\,|i,k\rangle = |k,i\rangle$. It has been shown in [96] that one may implement an approximate state evolution based on the walk operator $\mathcal{W}$ as follows:

$$\frac{1}{2}\mathcal{T}^{\dagger}(\mathcal{I}-j\mathcal{S})(j\mathcal{W})^{t\|\mathcal{H}\|_1}(\mathcal{I}+j\mathcal{S})\mathcal{T} = e^{-jt\mathcal{H}} + O(\theta t). \tag{2.34}$$

Recent contributions [11,96] have improved the error scaling in (2.34), which may also be viewed as reducing the overall computational complexity. Remarkably, the "quantum signal processing" technique proposed in [11] achieves a complexity on the order of $O(td\|\mathcal{H}\|_{\max} + \frac{\log 1/\epsilon}{\log\log 1/\epsilon})$, which is proved to be optimal in [97].

The Hamiltonian simulation problem plays a particularly important role in the complexity theory of quantum computation, since it is known to be **BQP-complete** [98,99]. To elaborate, BQP stands for Bounded-error Quantum Polynomial time, which marks the specific class of computational tasks that can be solved efficiently by quantum computers within a number of operations that scales polynomially with the size of the input data, at a success probability higher than 2/3 [100]. BQP-complete problems are such problems, that every BQP problem can be reduced to them using polynomial number of quantum operations. Intuitively, these problems may be viewed as the most representative and the most computationally challenging problems in the BQP class. To elaborate a little further, in computational complexity theory, "reducing problem A to problem B" refers to the process of transforming problem A to problem B using a certain amount of computational resources. For example, the problem of computing $5 \times 6 \times 7$ may be reduced to that of computing $30 \times 7$ using one operation (a multiplication). Intuitively, this toy example may be viewed as a "task addition" formula of $\mathcal{C}(5 \times 6 \times 7) = \mathcal{C}(5 \times 6) + \mathcal{C}(30 \times 7)$. Since any BQP problem can be reduced to BQP-complete problems using polynomial number of quantum operations (which is negligible in the sense that the BQP class is closed under the aforementioned "task addition" [101]), we may conclude that the computational class constituted by all BQP-complete problems are "the most representative problems" of the BQP class.

In practice, quantum algorithms do not necessarily fall completely into one of these categories. In fact, many practical quantum algorithms are conceived based on a beneficial fusion of subroutines of those different categories. A representative example of such algorithms is the Harrow-Hassidim-Lloyd (HHL) algorithm [16] aiming for solving linear systems having the form of $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, which is capable of providing exponential speedup over classical algorithms in solving some specific linear algebraic problems. Specifically, the HHL algorithm provides a high speedup, when the matrix $\boldsymbol{A}$ is well-conditioned, since the computational complexity is on the order of $O(\log N\kappa^2)$, where $N$ denotes the dimensionality of $\boldsymbol{x}$, and $\kappa$ is the condition number of $\boldsymbol{A}$.

The HHL algorithm is constituted by a successful blend of subroutines of all three categories. It relies substantially on a subroutine known as $\mathcal{U}_{\text{invert}}$ [16], which essentially

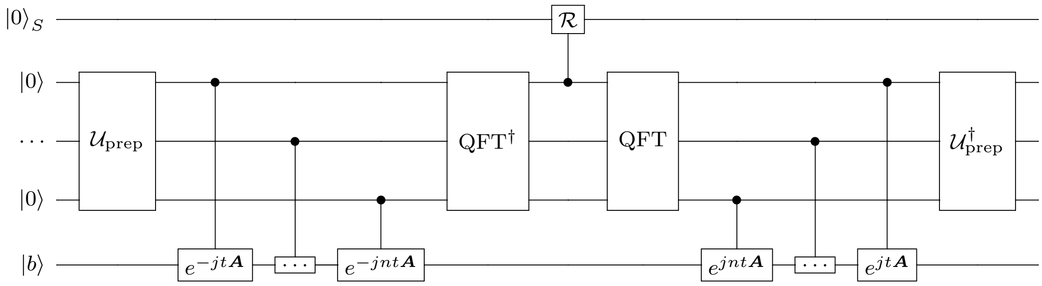


FIGURE 2.6: The $\mathcal{U}_{\text{invert}}$ subroutine in the HHL algorithm. The operation $\mathcal{R}$ represents the conditional rotation that takes the reciprocal of the eigenvalues. The operations $e^{-jkt\boldsymbol{A}}$, $k = -n \dots n$ may be implemented using Hamiltonian simulation.

inverts the matrix $\boldsymbol{A}$ with a small probability $p$ determined by the condition number of matrix $\boldsymbol{A}$ [16]. This is achieved by first computing the eigendecomposition of $\boldsymbol{A}$, then taking the reciprocal of all the "well-conditioned" eigenvalues (i.e., those that are larger than $1/(2\kappa)$), and finally apply the inverse operation (termed as "uncomputing" [16]) of the eigendecomposition to reconstruct $\boldsymbol{A}^{-1}$, as portrayed in Fig. 2.6. In particular, the eigendecomposition is implemented by computing $e^{-jt\boldsymbol{A}}$ for several different $t$ using Hamiltonian simulation, and then obtaining the eigenvalues of these operators — hence also that of $\boldsymbol{A}$ — using quantum phase estimation. Indeed, comparing Fig. 2.4 with Fig. 2.6, we see that the circuit in Fig. 2.6 before the controlled-$\mathcal{R}$ gate is in fact a quantum phase estimation circuit computing the eigendecomposition of $\boldsymbol{A}$. Moreover, the circuit after the controlled-$\mathcal{R}$ gate is exactly the inverse of the quantum phase estimation circuit, hence can reconstruct the matrix $\boldsymbol{A}^{-1}$ from its eigendecomposition, obtained by applying the controlled-$\mathcal{R}$ gate to the eigendecomposition of $\boldsymbol{A}$. The controlled-$\mathcal{R}$ gate implements the operation of taking the reciprocal of eigenvalues relying on the ancillary qubit $S$, which indicates whether the inversion is successful or not. Finally, the success probability $p$ is enhanced using an outer iteration of quantum amplitude amplification algorithm, which may be expressed as follows [16]:

$$\mathcal{U}_{\text{iter}} = \mathcal{U}_{\text{invert}}\mathcal{B}\mathcal{R}_{\text{init}}\mathcal{B}^\dagger\mathcal{U}_{\text{invert}}^\dagger\mathcal{R}_{\text{succ}}, \tag{2.35}$$

where $\mathcal{B}$ denotes the operator that computes the state $|b\rangle$ taking the initial state $|\psi\rangle_{\text{init}}$ as the input, and the operators $\mathcal{R}_{\text{init}}$ and $\mathcal{R}_{\text{succ}}$ represent the reflection operators of the quantum amplitude amplification algorithm, given by [16]

$$\begin{aligned} \mathcal{R}_{\text{init}} &= \mathcal{I} - 2\,|\psi\rangle_{\text{init}}\,\langle\psi|_{\text{init}}\,, \\ \mathcal{R}_{\text{succ}} &= \mathcal{I}_S - 2\,|1\rangle_S\,\langle 1|_S\,, \end{aligned} \tag{2.36}$$

respectively, where the subscript $S$ means that the operation is only executed on the ancilla $S$. Comparing (2.36) to both (2.25) and (2.24), we see that $\mathcal{R}_{\text{init}}$ corresponds to the reflection with respect to the initial state $|\psi\rangle_{\text{init}}$, while $\mathcal{R}_{\text{succ}}$ corresponds to the oracle.

### 2.5.2 Hybrid Quantum-Classical Computation

As discussed in Section 2.5.1, in many existing quantum algorithms, such as Shor's algorithm, the HHL algorithm, and in Hamiltonian simulation [11,12,16], the evaluation of eigenvalues and eigenvectors is a fundamental subroutine. In early contributions, quantum phase estimation [48, Sec. 5.2] was the default algorithm for eigenvalue evaluation, which typically requires a long coherence time. To enable eigenvalue evaluation on noisy intermediate-scale quantum computers, hybrid quantum-classical algorithms based on variational optimization have been proposed in [62,102,103].

Mathematically, for a Hermitian matrix $\boldsymbol{H}$, the eigenvector $\boldsymbol{\psi}_0$ corresponding to the smallest eigenvalue can be calculated as follows [62]

$$\boldsymbol{\psi}_{\mathrm{g}} = \arg\min_{\boldsymbol{\psi}} \ \boldsymbol{\psi}^{\dagger}\boldsymbol{H}\boldsymbol{\psi}, \tag{2.37a}$$

$$\text{subject to } \ \|\boldsymbol{\psi}\|_2 = 1. \tag{2.37b}$$

This problem is referred to as the variational formulation of eigenvalue evaluation in the literature [34, 62, 104]. If the eigenvectors are reparametrized using a vector $\boldsymbol{\theta}$, yielding $\boldsymbol{\psi} = \boldsymbol{\psi}(\boldsymbol{\theta})$, the task of finding the smallest eigenvalue or the corresponding eigenvector may be accomplished by searching for the minimum of the objective function [62]

$$J(\boldsymbol{\theta}) = \boldsymbol{\psi}^{\dagger}(\boldsymbol{\theta})\boldsymbol{H}\boldsymbol{\psi}(\boldsymbol{\theta}) \tag{2.38}$$

in the space of $\boldsymbol{\theta}$, while satisfying the normalization constraint (2.37b). The formulation (2.38) has been applied to the electronic structure computation of the hydrogen molecule[3] in [67]. In general, this would be a non-convex problem with respect to $\boldsymbol{\theta}$, which may be solved using iterative non-convex optimization solvers, such as the classic gradient descent and the Nelder-Mead simplex method [105]. In each iteration, the objective function or another function (e.g. the gradient) is first evaluated at a specific point in the parameter space, and then the parameters are updated according to the function values.

Under the framework of quantum computation, the problem (2.37) can be recast as [62]

$$|\psi_{\mathrm{g}}(\boldsymbol{\theta})\rangle = \arg\min_{|\psi(\boldsymbol{\theta})\rangle} \ \langle\psi(\boldsymbol{\theta})|\,\mathcal{H}\,|\psi(\boldsymbol{\theta})\rangle\,, \tag{2.39}$$

where $\mathcal{H}$ is a quantum observable [48, Sec. 2.2.5] representing the matrix $\boldsymbol{H}$, and the state $|\psi_{\mathrm{g}}(\boldsymbol{\theta})\rangle$ here corresponds to the ground state of $\mathcal{H}$. The quadratic form $J(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})|\,\mathcal{H}\,|\psi(\boldsymbol{\theta})\rangle$ can be viewed as the expectation value of the observable $\mathcal{H}$. The constraint (2.37b) is automatically satisfied due to the normalization property of quantum

---

[3] The problem of hydrogen molecular electronic structure computation aims to find the bond length corresponding to the ground state of the hydrogen molecule. In this case, the objective function $J(\theta)$ is the ground state energy, while the parameter $\theta$ (which is a scalar) corresponds to the inter-atom distance.

states. Taking decoherence into account, (2.39) can be reformulated equivalently for mixed states as

$$\rho_{\mathrm{g}}(\boldsymbol{\theta}) = \underset{\rho(\boldsymbol{\theta})}{\operatorname{argmin}} \ \operatorname{Tr}\{\mathcal{H}\rho(\boldsymbol{\theta})\}. \tag{2.40}$$

For conciseness of our discussion, we shall use the pure-state formulation (2.39) hereafter, whenever there is no confusion.

The essence of hybrid quantum-classical computation is to evaluate the functions using a quantum circuit, whereas the parameter values are updated using a classical computer, as illustrated in Fig. 1.1. To be more specific, a schematic of the quantum circuit evaluating the objective function $J(\boldsymbol{\theta})$ is portrayed in Fig. 2.7. The input state of the circuit is typically the all-zero state $|0\rangle^{\otimes n}$. The function-evaluation circuit $\mathcal{U}(\boldsymbol{\theta})$ encodes the parameter vector $\boldsymbol{\theta}$, and transforms the input state to the state $|\psi(\boldsymbol{\theta})\rangle$. The expectation value $J(\boldsymbol{\theta})$ is then computed based on the result of multiple measurements. This is achieved by decomposing the observable $\mathcal{H}$ (involving at most $K$-qubit interactions) as [62]

$$\mathcal{H} = \sum_{k=1}^{K} \left\{ \sum_{i_1, j_1, \ldots, i_k, j_k} h_{i_1, i_2, \ldots, i_k}^{(j_1, j_2, \ldots, j_k)} \prod_{l=1}^{k} \sigma_{i_l}^{(j_l)} \right\}, \tag{2.41}$$

where $\sigma_{i_l}^{(j_l)}$ denotes a Pauli-$j_l$ operator (i.e., $j_l$ may be $\mathcal{X}$, $\mathcal{Y}$ or $\mathcal{Z}$) acting on the $i_l$-th qubit. In light of this, the term $\prod_{l=1}^{k} \sigma_{i_l}^{(j_l)}$ can be implemented using a simple quantum circuit consisting of $k$ single-qubit gates followed by measurements, as shown in the dashed box of Fig. 2.7. For example, a Pauli-$\mathcal{Z}$ operator in (2.41) corresponds to a direct measurement, whereas a Pauli-$\mathcal{X}$ operator corresponds to a Hadamard gate followed by measurement. Thus, the expectation value can be obtained by measuring the outputs of these simple circuits, and evaluating a weighted sum over them using the weights $h_{i_1, i_2, \ldots, i_k}$. Note that this weight sum (represented by the operator $\mathcal{W}$ in Fig. 2.7) may be calculated using a classical computer.

In contrast to "fully quantum" algorithms (e.g. Shor's algorithm and Grover's search algorithm [106]) aiming to arrive at one of the computational basis states at the very end of computation, hybrid quantum-classical algorithms aim for computing the expectation values. Therefore, the measurement results have to be averaged over a number of independent circuit executions. In this thesis, we will refer to this process as "circuit sampling".

To portray the potential advantage of hybrid quantum-classical computation, we provide a rudimentary complexity comparison between classical computation and hybrid quantum-classical computation. Using classical computation, evaluating $J(\boldsymbol{\theta})$ for $\boldsymbol{\psi} \in \mathbb{C}^{2^n}$ requires on the order of $O(2^{2n})$ operations. By contrast, the complexity of the hybrid scheme depends both on the complexity of the function-evaluation circuit as well as on the structure of $\mathcal{H}$. More precisely, denoting the complexity of the function-evaluation circuit in terms of quantum gates as $T$, the total complexity of evaluating

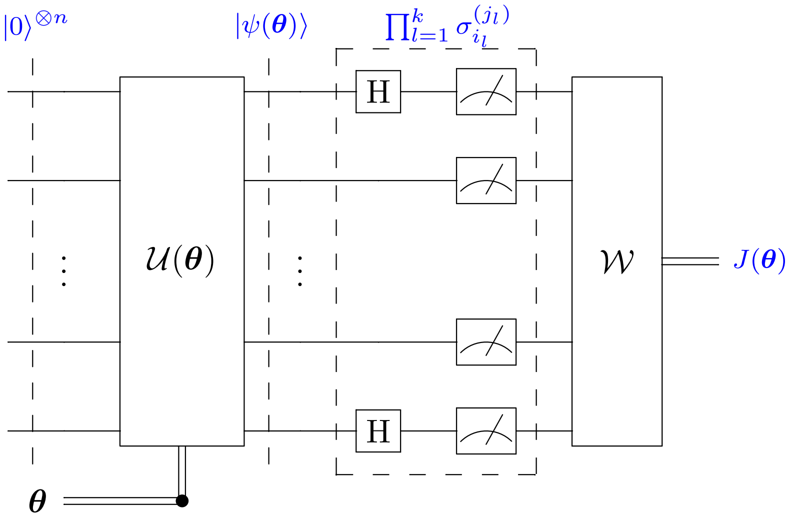


FIGURE 2.7: Portrayal of a quantum circuit evaluating the objective function $J(\boldsymbol{\theta})$ in a hybrid quantum-classical algorithm. The operator $\mathcal{W}$ represents the weighted averaging operation. The gates in the dashed box are chosen according to the observable decomposition (2.41).

$J(\boldsymbol{\theta})$ would be $O(T\prod_{k=1}^{K} N_k)$. Therefore, for application scenarios where the observable $\mathcal{H}$ is "sparse" in the sense that the number of terms $\prod_{k=1}^{K} N_k$ is small (e.g. polynomial in $n$), a substantial speedup over classical computation may be achieved, when using the hybrid approach.

# Chapter 3

# Introduction to Quantum Error Mitigation

In Chapter 2, we have introduced the fundamental units of quantum information processing, namely the qubits, and representative quantum algorithms based on gates (i.e. quantum operations) acting on qubits. In particular, we have discussed the family of quantum algorithms conceived under different error models, including "conventional" quantum algorithms based on error-free assumptions and hybrid quantum-classical algorithms aiming for potential applications to be run on near-term NISQ computers. In practice, errors are inevitable, while the error-free assumptions may only be approximately satisfied at the circuit-level using quantum error correction codes (QECCs). Against this background, the concept of quantum error mitigation (QEM) is proposed to meet the computational accuracy requirements of practical quantum algorithms executed on NISQ computers, by mitigating both the circuit-level errors and the algorithm-level errors.

In this chapter, we present a brief overview of the error sources in quantum computation as well as their countermeasures, with an emphasis on QEM techniques. The rest of this chapter is organized as follows. We first discuss the error sources in quantum computation in Section 3.1, including circuit-level error sources in Section 3.1.1 and algorithm-level error sources in Section 3.1.2. Next, in Section 3.2, we briefly introduce the conventional countermeasures against circuit-level errors, namely QECCs and QEDCs. We then discuss four main types of circuit-level QEM techniques in Section 3.3, followed by a discussion on algorithm-level QEM techniques in Section 3.4.

## 3.1 Error Sources in Quantum Computation

Errors are ubiquitous in practical computational systems. At the top level of a computational system, tasks are performed based on algorithms, which may be probabilistic or approximate in nature, hence result in computational errors. At a lower level of the system, algorithms are eventually executed relying on physical components such as circuits or mechanical structures. These components may themselves be imperfect, i.e., affected by noise, which may accumulate through the entire process of computation and lead to errors in the final result.

### 3.1.1 Circuit-Level Error Sources

At the circuit level, quantum algorithms are compiled into gates. Ideal quantum gates can be modelled as unitary operators acting on quantum states. However, due to the imperfection of the gate implementation or external noises, the actual operations applied to the quantum states are typically non-unitary. As discussed in Section 2.4, the undesired operations play the role of imperfect "quantum channels", which are formally modelled as CPTP operators.

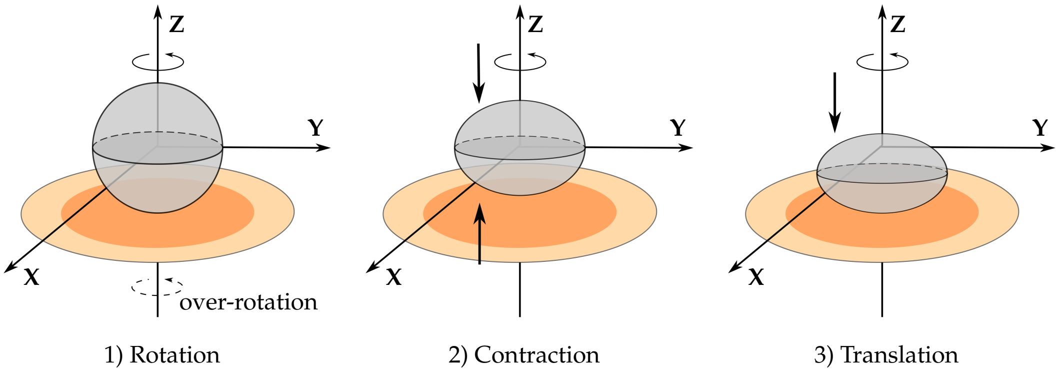
To be more specific, let us consider some tangible examples of single-qubit quantum channels. According to Section 2.4, the effect of a single-qubit quantum channel $\boldsymbol{C}$ on the state $x$ may be viewed as an affine transformation, consisting of a linear transformation and a translation. Moreover, according to our discussion in Section 2.3, the space containing all single-qubit quantum states may be geometrically represented by the Bloch sphere. Therefore, the effect of a quantum channel may be interpreted as an undesired affine transformation applied to the Bloch sphere,[1] potentially including rotation, contraction (dilation is not allowed by the trace-preserving condition [48, Theorem 9.2]) and translation, as portrayed in Fig. 3.1. By contrast, an ideal unitary gate only corresponds to a pure rotation of the Bloch sphere.

Pauli channels constitute a class of single-qubit channels having paramount significance, which can be represented by diagonal matrices under the PTM formulation [107]. Geometrically, Pauli channels may be interpreted as pure contractions to the Bloch sphere. Under the operator-sum representation, a general Pauli channel may be expressed as

$$\mathcal{C}_{\text{Pauli}}(\rho) = (1 - p_{\text{X}} - p_{\text{Y}} - p_{\text{Z}})\rho + p_{\text{X}}\boldsymbol{S}_{\mathcal{X}}\rho\boldsymbol{S}_{\mathcal{X}} + p_{\text{Y}}\boldsymbol{S}_{\mathcal{Y}}\rho\boldsymbol{S}_{\mathcal{Y}} + p_{\text{Z}}\boldsymbol{S}_{\mathcal{Z}}\rho\boldsymbol{S}_{\mathcal{Z}}. \tag{3.1}$$

---

[1] Strictly speaking, quantum channels only apply to quantum states. However, in order to gain further insights, we may alternatively view the actions of quantum channels on the collection of all quantum states as affine transformations applied to the entire space of all quantum states, geometrically represented as the Bloch sphere.



1) Rotation 2) Contraction 3) Translation

FIGURE 3.1: Geometrical interpretation of an imperfect quantum gate acting on a qubit: Sequential application of 1) rotation 2) contraction and 3) translation to the Bloch sphere. The ideal gate is a rotation around the Z-axis.

As special cases, when $p_Y = p_Z = 0$, the channel is termed as a **bit-flip channel**, also known as an X-error channel. Similarly, Pauli channels corresponding to $p_X = p_Y = 0$ are termed as **phase-flip channels**, also known as Z-error channels. The channels satisfying $p_X = p_Y = p_Z = p \neq 0$ are termed as **depolarizing channels**. These specific Pauli channels are particularly important both in practice and in the analysis of quantum error correction codes, which will be further elaborated on in Section 3.2. Geometrically, bit-flip channels correspond to contractions towards the X-axis, phase-filp channels correspond to contractions towards the Z-axis, while depolarizing channels correspond to homogeneous contractions towards the center of the Bloch sphere.

When the quantum channel consists of pure rotations, it is termed as a **coherent channel** [108], sometimes also referred to as an over-rotation channel [38], because the ideal gate itself is the "exact rotation". Compared to Pauli channels, coherent channels do not necessarily incur information loss, since they only involve basis transformations, but may still cause deleterious effects if the channel is not known before activating the quantum circuit. Especially, coherent channels may even be more detrimental than Pauli channels for certain quantum error correction codes [109].

A quantum channel frequently encountered by practical quantum computers is the **amplitude damping channel** [48, Section 8.3.5], which has the following operator-sum representation

$$\mathcal{C}_{\mathrm{damp}}(\rho) = \boldsymbol{E}_0 \rho \boldsymbol{E}_0^{\dagger} + \boldsymbol{E}_1 \rho \boldsymbol{E}_1^{\dagger}, \tag{3.2}$$

where

$$\boldsymbol{E}_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix}, \ \boldsymbol{E}_1 = \begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix}, \tag{3.3}$$

with $\gamma$ termed as the amplitude damping probability. The amplitude damping channel is an abstraction of the ubiquitous phenomenon of undesired energy dissipation, for example, the loss of a photon. This may be recognized by considering its effect on the computational basis states, as follows:

$$\mathcal{C}_{\text{damp}}(|0\rangle\langle 0|) = |0\rangle\langle 0|\,,\ \ \mathcal{C}_{\text{damp}}(|1\rangle\langle 1|) = (1-\gamma)\,|1\rangle\langle 1| + \gamma\,|0\rangle\langle 0|\,. \tag{3.4}$$

In practical quantum computers, the state $|0\rangle$ typically refers to the "ground state" having the lowest energy, while the state $|1\rangle$ is an excited state having higher energy [48, Section 8.3.5]. Hence from (3.4) we observe that with probability $\gamma$, the amplitude damping channel would incur an energy loss. Geometrically, amplitude damping channels correspond to contractions towards the northpole of the Bloch sphere, namely the $|0\rangle$ state. The PTM of amplitude damping channels is given by

$$\boldsymbol{C}_{\text{damp}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{1-\gamma} & 0 & 0 \\ 0 & 0 & \sqrt{1-\gamma} & 0 \\ \gamma & 0 & 0 & 1-\gamma \end{bmatrix}. \tag{3.5}$$

Observe from (3.5) that amplitude damping channels may be alternatively viewed as the composition of a contraction towards the center of the sphere (i.e., a depolarizing channel), a contraction towards the Z-axis (i.e., a bit-flip channel), and a translation towards the northpole $|0\rangle$.

### 3.1.2 Algorithm-Level Error Sources

Most existing quantum algorithms are essentially probabilistic. This originates from the fact that the output states of quantum algorithms are in general not the eigenstates of the observables, and thus the subsequent measurement outcomes are not deterministic. For example, in the Grover's popular search algorithm [13] conceived for unstructured search over $N$ database entries, the probability of success after $k$ iterations is given by [13]

$$p_{\text{success}} = \sin^2\left[2\left(k+\frac{1}{2}\right)\arcsin\left(\frac{1}{\sqrt{N}}\right)\right], \tag{3.6}$$

which might not reach exactly 1 for any integer $k$. Similarly, Shor's factoring algorithm [12], which is one of the few quantum algorithms shown to provide exponential speedup over their classical counterparts, also has a constant probability of failure. Due to the probabilistic nature of quantum algorithms, the "polynomial time" complexity class for quantum computers is termed as Bounded-error Quantum Polynomial time (BQP) [100]. This implies that there is a fundamental trade-off between the complexity and the accuracy of quantum computation, since a near-unity successful probability is achievable through multiple repetitions of an algorithm having bounded error probability.

Apart from the above-mentioned probabilistic errors, many practical quantum algorithms also suffer from approximation errors. A representative example is constituted by Hamiltonian simulation [11], which aims for performing a state evolution according to the exponentiation $\exp\{j\tau\mathcal{H}\}$ of the Hamiltonian $\mathcal{H}$, relying on a linear decomposition of $\mathcal{H}$ as

$$\mathcal{H} = \sum_{k=1}^{K} \theta_k \mathcal{H}_k, \tag{3.7}$$

where for each $\mathcal{H}_k$, the corresponding exponentiation $\exp\{j\tau\theta_k\mathcal{H}_k\}$ can be readily carried out by native gates in the quantum computer. However, the implementation of the global evolution $\exp\{j\tau\mathcal{H}\}$ is not straightforward, since the components $\mathcal{H}_k$, $k = 1 \ldots K$ do not in general commute with each other. To this end, the Trotterization technique (also known as the Lie-Trotter product formula) [75] is usually applied, which divides the entire state evolution duration $\tau$ into various "Trotter steps" as follows

$$\begin{aligned} e^{j\tau\mathcal{H}} &= \left(e^{j\tau/N_\tau\mathcal{H}}\right)^{N_\tau} \\ &= \left\{\left(e^{j\frac{\tau}{2N_\tau}\mathcal{H}_1} \ldots e^{j\frac{\tau}{2N_\tau}\mathcal{H}_K}\right)\left(e^{j\frac{\tau}{2N_\tau}\mathcal{H}_K} \ldots e^{j\frac{\tau}{2N_\tau}\mathcal{H}_1}\right)\right\}^{N_\tau} + O(N_\tau^{-2}), \end{aligned} \tag{3.8}$$

where the terms on the order of $O(N_\tau^{-2})$ are neglected in practice, hence result in an approximation error. The formula (3.8) is termed as the second-order Trotterization. Another popular variant of Trotterization, also known as the first-order Trotterization, which is given by [75]

$$e^{j\tau\mathcal{H}} = \left(\prod_{k=1}^{K} e^{j\tau/N_\tau\mathcal{H}_k}\right)^{N_\tau} + O(N_\tau^{-1}), \tag{3.9}$$

that has an error on the order of $O(N_\tau^{-1})$.

Naturally, probabilistic errors and approximation errors may be mitigated by increasing the number of circuit activations (i.e., the number of samples) and the depth of the circuit (e.g., the number of Trotter steps), respectively. However, these naive approaches may be computationally inefficient, and sometimes outright unaffordable. For example, the achievable depth of quantum circuits on NISQ computers is limited by their coherence time, and thus one cannot simply mitigate the error due to Trotterization by increasing the number of Trotter steps. This motivates the development of algorithm-level QEM methods, which will be discussed in more detail in Section 3.4.

## 3.2 Quantum Error Correction/Detection Codes

In quantum computation, by far the most plausible technique conceived for combatting circuit-level errors is quantum error correction codes (QECCs) [24, 52]. Conceptually, QECCs bear resemblance with classical error correction codes relying on syndrome-based decoding. Classically, for a $(n, k)$ linear block code, redundancy is injected into the transmitted codeword using a $k \times n$ generator matrix $\boldsymbol{G} \in \mathbb{F}_2^{k\times n}$ formulated as

$$\boldsymbol{y}^{\mathrm{T}} = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{G}, \tag{3.10}$$

where we have $\boldsymbol{x} \in \mathbb{F}_2^k$ and $\boldsymbol{y} \in \mathbb{F}_2^n$, and the matrix multiplication is carried out over the binary finite field $\mathbb{F}_2$. Apparently, a legitimate codeword $\boldsymbol{y}$ would reside in the row space of $\boldsymbol{G}$, hence it is orthogonal to the left null space of $\boldsymbol{G}$. In other words, given a matrix $\boldsymbol{H}$ whose columns form a basis of the left null space of $\boldsymbol{G}$, we should have

$$\boldsymbol{y}^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}} = \boldsymbol{0}, \tag{3.11}$$

which holds for every legitimate codeword $\boldsymbol{y}$. The matrix $\boldsymbol{H}$, termed as the parity check matrix (PCM), may be formulated for a so-called "systematic code" as

$$\boldsymbol{G} = [\boldsymbol{I}_k|\boldsymbol{P}], \;\; \boldsymbol{H} = [\boldsymbol{P}^{\mathrm{T}}|\boldsymbol{I}_{n-k}], \tag{3.12}$$

which reflects that the identity matrix part directly copies the original information bits into the encoded word, while the parity part represents the redundant bits providing the correction capability. When the received word is corrupted by an additive error vector $\boldsymbol{e}$ as $\boldsymbol{r} = \boldsymbol{y} + \boldsymbol{e}$, with $\boldsymbol{e}$ containing a binary 1 in the particular position, where the legitimate codeword was corrupted, one may obtain the **error syndrome** as follows

$$\begin{aligned} \boldsymbol{s}^{\mathrm{T}} &= \boldsymbol{r}^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}} \\ &= \boldsymbol{e}^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}}, \end{aligned} \tag{3.13}$$

where the second line follows from $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}} = \boldsymbol{0}$. The hard-decision-based maximum likelihood decoder may then be realized by constructing a look-up table containing all legitimate syndrome vectors $\boldsymbol{s}$, each associated with a correctable $\boldsymbol{e}$, having the lowest Hamming weight.

A quantum-domain counterpart of the error syndrome, namely the stabilizers [52], may be applied to identify the circuit-level errors in quantum states. Specifically, a stabilizer $\mathcal{G}$ of the quantum state $|\psi\rangle$ satisfies

$$\mathcal{G}\,|\psi\rangle = |\psi\rangle\,. \tag{3.14}$$

In other words, the state $|\psi\rangle$ is the eigenvector of $\mathcal{S}$ associated with the eigenvalue $+1$. We require furthermore that the stabilizer $\mathcal{G}$ should be a Hermitian operator, ensuring that it is observable. In general, a state $|\psi\rangle$ might have multiple stabilizers. This formalism is motivated by the fact that, when we measure the stabilizer $\mathcal{G}$ for a quantum state satisfying (3.14), we would be informed that an error has occurred, if we obtain an outcome of $-1$.[2] Furthermore, if we obtain $+1$ from the measurement, the original state $|\psi\rangle$ is unchanged due to (3.14).

The error-correction capability of QECCs hinges on the specific design of a **stabilizer group** $\mathscr{S}$, which contains a set of stabilizers $\{\mathcal{G}_i\}_{i=1}^{K}$ obeying the commutative property $[\mathcal{G}_i, \mathcal{G}_j] = 0$, $\forall i \neq j$, $\mathcal{G}_i, \mathcal{G}_j \in \mathscr{G}$. In particular, we have

$$\mathcal{G}_i |\psi\rangle = \mathcal{G}_j |\psi\rangle = \mathcal{G}_i \mathcal{G}_j |\psi\rangle . \tag{3.15}$$

Based on the specific terminology that will be detailed in Chapter 7, (3.15) implies that the elements in the stabilizer group are **simultaneously observable**. This means that their error-detection capability can be exploited simultaneously for the state $|\psi\rangle$. We may better understand the error-detection property of stabilizers by considering their effect when applied to erroneous states. For example, if the state $|\psi\rangle$ is contaminated by an error operator $\mathcal{P}$, we obtain the erroneous state $|\varphi\rangle = \mathcal{P}|\psi\rangle$. The outcome of the stabilizer $\mathcal{G}$ of $|\psi\rangle$ acting upon $|\varphi\rangle$ may then be expressed as

$$\mathcal{G}|\varphi\rangle = \mathcal{G}\mathcal{P}|\psi\rangle = \begin{cases} |\varphi\rangle , & [\mathcal{G}, \mathcal{P}] = 0; \\ -|\varphi\rangle , & \{\mathcal{G}, \mathcal{P}\} = 0. \end{cases} \tag{3.16}$$

The conditions (3.16) are particularly useful when the stabilizer $\mathcal{S}$ belongs to the Pauli group $\mathscr{P}$. To elaborate a little further, a Pauli operator $\mathcal{A} \in \mathscr{P}$ would either commute with another Pauli operator $\mathcal{B}$, i.e. $[\mathcal{A}, \mathcal{B}] = 0$, or anti-commute with $\mathcal{B}$, i.e. $\{\mathcal{A}, \mathcal{B}\} = 0$. When the stabilizer $\mathcal{G}$ itself is a Pauli operator, once we measure $\mathcal{G}$ with respect to a state $|\varphi\rangle$, $|\varphi\rangle$ would automatically collapse to a state corresponding to either the $+1$ eigenvalue or the $-1$ eigenvalue. According to (3.16), we see that the collapsed state, denoted as $|\tilde{\varphi}\rangle$, corresponds to an error operator $\tilde{\mathcal{P}}$ in the Pauli group. In other words, regardless of what the original error $\mathcal{P}$ is, it would become a Pauli error, once we measure a Pauli stabilizer $\mathcal{G}$. This phenomenon illustrates the importance of Pauli channels, since a stabilizer check may be viewed as "Pauli twirling" capable of transforming an arbitrary quantum channel into a Pauli channel [110].

We can now see that the stabilizer group may be practically constructed using Pauli operators, we may then consider a mapping from the single-qubit Pauli operators to integers as follows:

$$\boldsymbol{I} \rightarrow 0,\ \boldsymbol{X} \rightarrow 1,\ \boldsymbol{Z} \rightarrow 2,\ \boldsymbol{Y} \rightarrow 3, \tag{3.17}$$

which may be represented alternatively in a binary form as

$$\boldsymbol{I} \rightarrow 00,\ \boldsymbol{X} \rightarrow 01,\ \boldsymbol{Z} \rightarrow 10,\ \boldsymbol{Y} \rightarrow 11. \tag{3.18}$$

---

[2] Note that the only possible outcomes are $+1$ and $-1$, since $\mathcal{G}$ is both unitary and Hermitian.

The mapping (3.18) is also known as the **Pauli-to-binary isomorphism** [111], which is an isomorphism from the Pauli group to GF(2) × GF(2). Correspondingly, the multiplication operation in the Pauli group is mapped to the addition operation on GF(2) × GF(2) (up to a global phase), as may be verified by considering the examples in (3.18), and omitting the global phase by treating $\boldsymbol{Y}$ as $\boldsymbol{Y} = \boldsymbol{XZ} = \boldsymbol{ZX}$.[3] In light of this isomorphism, we may view a stabilizer group as a quaternary linear block code, usually termed as a quantum stabilizer code (QSC) [24,52]. The quaternary code may be further decomposed into a pair of binary linear block codes having the PCMs $\boldsymbol{H}_\mathrm{x}$ and $\boldsymbol{H}_\mathrm{z}$, respectively. This is plausible, since we have $\boldsymbol{e}^\mathrm{T}\boldsymbol{H}^\mathrm{T} = \boldsymbol{0}$ when $\boldsymbol{e}$ is an undetectable error for a classical linear block code. By contrast, we would have $\tilde{\boldsymbol{e}}^\mathrm{T}\tilde{\boldsymbol{H}}^\mathrm{T} = \boldsymbol{0}$, where $\tilde{\boldsymbol{H}} = [\boldsymbol{H}_\mathrm{x}|\boldsymbol{H}_\mathrm{z}]$ is the PCM of the entire QSC, if $\tilde{\boldsymbol{e}}$ is an undetectable error of the stabilizer group (i.e. equivalent to the identity operator $\mathcal{I}$ within the stabilizer group). QSCs composed of such a pair of codes are termed as Calderbank-Shor-Steane (CSS) codes [41,112], which constitute a large class of practical QSCs whose members are eminently suitable for quantum computation. An additional constraint termed as the "symplectic criterion" should be satisfied by CSS codes, in order to preserve the commutativity properties between different Pauli operators [24], which is given by

$$\boldsymbol{H}_\mathrm{z}\boldsymbol{H}_\mathrm{x}^\mathrm{T} + \boldsymbol{H}_\mathrm{x}\boldsymbol{H}_\mathrm{z}^\mathrm{T} = \boldsymbol{0}, \tag{3.19}$$

where the "+" operation should be interpreted as the addition operation on GF(2).

To provide further intuitions concerning the construction of QSCs, let us consider a specific example, namely Steane's 7-qubit code [41]. The stabilizers of the 7-qubit codes are as follows:

$$\begin{aligned}&\mathcal{G}_1 = \boldsymbol{I}_1\boldsymbol{I}_2\boldsymbol{I}_3\boldsymbol{X}_4\boldsymbol{X}_5\boldsymbol{X}_6\boldsymbol{X}_7,\ \mathcal{G}_2 = \boldsymbol{I}_1\boldsymbol{X}_2\boldsymbol{X}_3\boldsymbol{I}_4\boldsymbol{I}_5\boldsymbol{X}_6\boldsymbol{X}_7,\ \mathcal{G}_3 = \boldsymbol{X}_1\boldsymbol{I}_2\boldsymbol{X}_3\boldsymbol{I}_4\boldsymbol{X}_5\boldsymbol{I}_6\boldsymbol{X}_7,\\ &\mathcal{G}_4 = \boldsymbol{I}_1\boldsymbol{I}_2\boldsymbol{I}_3\boldsymbol{Z}_4\boldsymbol{Z}_5\boldsymbol{Z}_6\boldsymbol{Z}_7,\ \mathcal{G}_5 = \boldsymbol{I}_1\boldsymbol{Z}_2\boldsymbol{Z}_3\boldsymbol{I}_4\boldsymbol{I}_5\boldsymbol{Z}_6\boldsymbol{Z}_7,\ \mathcal{G}_6 = \boldsymbol{Z}_1\boldsymbol{I}_2\boldsymbol{Z}_3\boldsymbol{I}_4\boldsymbol{Z}_5\boldsymbol{I}_6\boldsymbol{Z}_7,\end{aligned} \tag{3.20}$$

where $\boldsymbol{I}_i$, $\boldsymbol{X}_i$ and $\boldsymbol{Z}_i$ represent the identity, the Pauli-X, and the Pauli-Z operator acting upon the $i$-th qubit, respectively. According to the Pauli-to-binary isomorphism, we have an equivalent linear block code representation for this 7-qubit code, characterized by its PCM $\tilde{\boldsymbol{H}} = [\boldsymbol{H}_\mathrm{x}|\boldsymbol{H}_\mathrm{z}]$, where

$$\boldsymbol{H}_\mathrm{x} = \boldsymbol{H}_\mathrm{z} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1\\ 0 & 1 & 1 & 0 & 0 & 1 & 1\\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}, \tag{3.21}$$

obtained by replacing the Pauli-X operators (resp. Pauli-Z operators) in (3.20) with 1, while replacing the identity operators with 0. It is also straightforward to verify that (3.19) holds for this code.

---

[3] Omitting the global phase is plausible, since the action of a phase-shifted Pauli operator $e^{j\theta}\boldsymbol{P}$ on a specific quantum state $\rho$ may be expressed as $(e^{j\theta}\boldsymbol{P})\rho(e^{j\theta}\boldsymbol{P})^\dagger = \boldsymbol{P}\rho\boldsymbol{P}^\dagger$, where the global phase is cancelled out.

Depending on the error detection capability of a QSC, it may be referred to as either a quantum error correction code (QECC) or a quantum error detection code (QEDC). In particular, a CSS code composed of an $(n, k_1, d_1)$ code and an $(n, k_2, d_2)$ code $(k_1 > k_2)$ is referred to as a $[[n, k_1 - k_2, d]]$ code, where $d = \min\{d_1, d_2\}$ is the minimum distance of this CSS code. Similar to classical linear block codes, CSS codes having a minimum distance of $d$ can detect an arbitrary error pattern involving $(d-1)$ qubits, but only those involving no more than $(d-1)/2$ qubits can be corrected [113]. Therefore, CSS codes having a minimum distance of $d = 2$ are called QEDCs, since they can only detect a single error, but they are not capable of correcting any errors. Those QSCs that are capable of correcting at least one error are typically referred to as QECCs.

Naturally, the decoding of a QECC may be implemented using the decoding algorithms designed for the associated classical linear block code. This becomes more convenient when we consider CSS codes, since the pair of component codes are both binary codes, which have been studied extensively in classical coding theory. For QEDCs, once we detect an error, we may avoid further errors in the subsequent computation by simply discarding the circuit execution, and restart the entire process.

The theoretical significance of QECCs lies in the fact that **fault-tolerant quantum computation** may be achieved by concatenating multiple QECCs [48, Section 10.6.1], or using a sufficiently high block length for some specific classes of QECCs, for example, the topological codes [59, 114, 115]. Fault tolerance implies that the overall error probability can be reduced to an arbitrarily low level, but only at the cost of a computational overhead (specifically, qubit overhead for QECCs) increasing modestly with the number of gates. This ensures that the additional computational cost due to error correction does not completely erode the speed advantage of quantum computation. It is widely believed that fault tolerance is required for many practical quantum algorithms providing substantial quantum speedup, such as Shor's factoring algorithm and Grover's quantum search algorithm, since they rely on deep quantum circuits that would be extremely noisy without QECCs. By contrast, QEDCs may not be suitable candidates for achieving fault tolerance, since their additional computational cost due to the "discard-and-restart" process grows exponentially with the number of gates.

Sometimes a QECC may have a better error-correction performance than a classical error correction code having the same minimum distance $d$, due to its **degeneracy** [116]. Broadly speaking, degeneracy means that some undetectable non-zero error patterns may not necessarily degrade the computational result, hence they actually improve the error-correction capability. Formally, degeneracy refers to the phenomenon that some non-identity error operator commutes with all stabilizers. Note that degeneracy is a unique characteristic of QECCs, since all undetected error patterns are harmful for classical error correction codes.

Although the error-correction performance of QECCs has strong theoretical guarantees, using solely QECCs in its own right may not be sufficient for improving the accuracy of quantum computation, especially for state-of-the-art NISQ computers. To elaborate, although the computational overhead of error correction grows slowly with respect to the size of the circuit in an asymptotic regime, the overhead is still unaffordable for NISQ computers typically having only around 100 qubits and gate error rates in the range of $10^{-3} \sim 10^{-2}$ [31]. Furthermore, existing NISQ computers typically do not have qubit array topologies that are suitable for QECCs. Consequently, error correction and stabilizer measurements would require additional processing steps for swapping different qubits, which further exacerbates the computational overhead [117]. Moreover, QECCs are only effective in reducing circuit-level errors, since algorithm-level errors cannot be identified using stabilizer checks.

The limited capability of QECCs motivates the development of quantum error mitigation (QEM), which aims for mitigating the deleterious effect of both circuit-level and algorithm-level errors for quantum algorithms relying on short-depth circuits, at a low computational overhead. Typically, the computational overhead of QEM is much lower than that of QECCs for small-scale quantum circuits, but the overhead increases exponentially as the circuit size increases. Therefore, QEM is not a substitute of QECCs, but should rather be considered as a technique tailored for NISQ computers, and might be used in conjunction with QECCs for their mutual benefits. We will introduce some representative QEM method in Sections 3.3 and 3.4.

## 3.3 Circuit-Level Quantum Error Mitigation

In this section, we present a brief overview of the representative techniques suitable for circuit-level QEM [36, 37, 37–39, 64–66, 118–123]. The basic characteristics of these techniques, including their main computational overhead and the requirements concerning any prior knowledge about the computational task or the noise model, are shown in Table 3.1. In particular, the methods shown in bold fonts represent the main subject of study in the rest of this treatise. We will present our detailed performance analysis as well as generalized or improved versions of these techniques in Chapters 4, 5, 6 and 7.

### 3.3.1 Channel Inversion Based QEM

When contaminated by decoherence, the quantum circuits of hybrid quantum-classical computation would produce erroneous expectation values. Fortunately, the weighted-averaging nature of hybrid quantum-classical computation facilitates the conception of a qubit-overhead-free method that mitigates the deviation from the true expectation value, namely the channel inversion based QEM (CI-QEM). In this section, we

| | **Main overhead** | **Prior knowledge required** | **Remark** |
|---|---|---|---|
| Zero-noise extrapolation [37, 38, 118] | Sampling overhead | No | Requires pulse-level control, or less effective to errors that do not commute with the gates |
| **Channel inversion** [37, 39, 119] | Sampling overhead | Channel estimation (gate set tomography) | Has error floor due to imperfect channel estimation |
| Learning-based [64, 120, 121] | Sampling overhead | Pre-training on certain circuit sets | |
| **Symmetry verification** [36, 122] | Sampling overhead,<br>qubit overhead | Type of symmetries in the computational task | **Symmetry-based** |
| **Virtual distillation** [65, 66] | Sampling overhead,<br>qubit overhead | No | **Symmetry-based**;<br>Has error floor due to mismatch in the dominant eigenvector [123] |

TABLE 3.1: Comparisons between different circuit-level QEM methods. Methods in bold red fonts are the main subjects of study in this treatise.

introduce the formulation of CI-QEM and the computational overhead it incurs – the sampling overhead.

#### 3.3.1.1 The Basic Formulation of CI-QEM

The philosophy of CI-QEM is to insert a probabilistic quantum circuit right after every quantum gate, which reverts the effect of the quantum channel modelling the imperfection inflicted by the gate. Conceptually, a CI-QEM-protected gate can be portrayed as in Fig. 3.2. The imperfect gate (in this case an imperfect CNOT gate) can be decomposed into a perfect gate and a quantum channel $\mathcal{C}$. Given an input state having the density matrix $\rho_{\text{in}}$, according to (2.12), the output state after the imperfect gate is given by

$$\rho_{\text{out}} = \sum_i \boldsymbol{K}_i \boldsymbol{U}_{\text{g}} \rho_{\text{in}} \boldsymbol{U}_{\text{g}}^\dagger \boldsymbol{K}_i^\dagger, \tag{3.22}$$

where the matrix $\boldsymbol{U}_{\text{g}}$ corresponds to the effect of the perfect gate. In a vectorized form, the output state can be expressed as

$$\boldsymbol{x}_{\text{out}} = \boldsymbol{C}\boldsymbol{G}\boldsymbol{x}_{\text{in}}, \tag{3.23}$$

where we have $\boldsymbol{G} = \boldsymbol{U}_{\text{g}}^* \otimes \boldsymbol{U}_{\text{g}}$, and the Pauli transfer matrix $\boldsymbol{C}$ is given in (2.16). If we have an estimate $\hat{\boldsymbol{C}}$ of $\boldsymbol{C}$, potentially obtained using methods such as quantum process tomography [124], an estimate of the output of the decoherence-free gate $\boldsymbol{G}$ can be obtained as

$$\hat{\boldsymbol{x}} = \boldsymbol{M}\boldsymbol{x}_{\text{out}} = \hat{\boldsymbol{C}}^{-1}\boldsymbol{C}\boldsymbol{G}\boldsymbol{x}_{\text{in}}, \tag{3.24}$$

where $\boldsymbol{M} = \hat{\boldsymbol{C}}^{-1}$ is the Pauli transfer matrix representation of the probabilistic quantum circuit $\mathcal{M}$ constructed for inverting the channel.

To elaborate further, if a gate is followed directly by measurement, $\mathcal{M}$ is implemented by applying different circuits according to a probability distribution in different circuit executions and performing a weighted averaging over the measurement outcomes. This can be formulated as

$$\sum_{k=1}^{K} \boldsymbol{m}_k^{\mathrm{T}} \boldsymbol{M} \boldsymbol{x}_{\mathrm{out}} = \sum_{k=1}^{K} w_k p_k \boldsymbol{m}_k^{\mathrm{T}} \boldsymbol{M}_k \boldsymbol{x}_{\mathrm{out}}, \tag{3.25}$$

where $\boldsymbol{M}_k$ is the $k$-th candidate circuit applied at a probability of $p_k$, $w_k$ is the weight of the $k$-th potential measurement outcome, and $\boldsymbol{m}_k$ is the Pauli transfer matrix representation of the measurement operator corresponding to the $k$-th outcome. For a circuit constructed by multiple gates, the weights and probability distributions follow directed by linearity. For example, for a simple circuit containing two consecutive imperfect gates $\tilde{\boldsymbol{G}}^{(1)}$ and $\tilde{\boldsymbol{G}}^{(2)}$, we have

$$\begin{aligned} &\sum_{j=1}^{J}\sum_{k=1}^{K} \boldsymbol{m}_{jk}^{\mathrm{T}} \boldsymbol{M}^{(2)} \tilde{\boldsymbol{G}}^{(2)} \boldsymbol{M}^{(1)} \tilde{\boldsymbol{G}}^{(1)} \boldsymbol{x}_{\mathrm{in}} \\ &= \sum_{j=1}^{J}\sum_{k=1}^{K} w_j^{(1)} w_k^{(2)} p_j^{(1)} p_k^{(2)} \boldsymbol{m}_{jk}^{\mathrm{T}} \boldsymbol{M}_k^{(2)} \tilde{\boldsymbol{G}}^{(2)} \boldsymbol{M}_j^{(1)} \tilde{\boldsymbol{G}}^{(1)} \boldsymbol{x}_{\mathrm{in}}, \end{aligned} \tag{3.26}$$

where the superscripts "(1)" and "(2)" indicate the first and the second gate, respectively.

Since the expectation evaluation in hybrid quantum-classical computation is implemented by applying a linear transformation (weighted averaging) to the measurement outcomes, it fits nicely with CI-QEM. Both the numerical and experimental results of [38, 39] show that QEM is indeed capable of reducing the computational error in VQAs in the context of quantum chemistry problems. By contrast, "fully quantum" algorithms such as Shor's algorithm and Grover's algorithm operate in different ways, hence they might not be protectable by CI-QEM.

Optionally, one may apply a quantum channel precoder to the imperfect gate, yielding

$$\begin{aligned} \hat{\boldsymbol{x}}_{\mathrm{P}} &= \boldsymbol{M}_{\mathrm{P}} \boldsymbol{P}_2 \boldsymbol{C} \boldsymbol{G} \boldsymbol{P}_1 \boldsymbol{x}_{\mathrm{in}} \\ &= (\boldsymbol{P}_2 \hat{\boldsymbol{C}} \boldsymbol{G} \boldsymbol{P}_1 \boldsymbol{G}^{-1})^{-1} \boldsymbol{P}_2 \boldsymbol{C} \boldsymbol{G} \boldsymbol{P}_1 \boldsymbol{x}_{\mathrm{in}}, \end{aligned} \tag{3.27}$$

as shown in Fig. 3.2. The quantum channel precoder turns the channel $\boldsymbol{C}$ into another (possibly more preferable) channel $\boldsymbol{P}_2 \boldsymbol{C} \boldsymbol{G} \boldsymbol{P}_1 \boldsymbol{G}^{-1}$. For example, the so-called Pauli twirling of [38, 39, 125] may be viewed as a quantum channel precoder turning an arbitrary channel into a Pauli channel. Similarly, Clifford twirling [126] turns an arbitrary channel into a depolarizing channel. According to the operator-sum representation [48, Sec. 8.2.4], a quantum channel precoder can be implemented by a probabilistic mixture

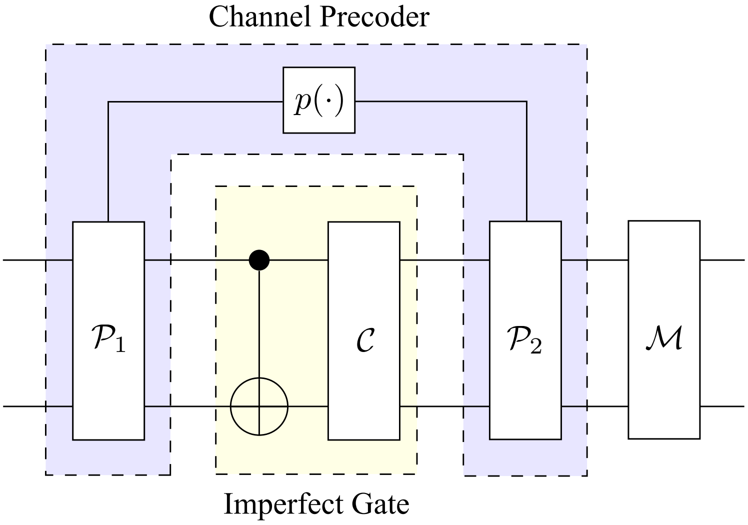


FIGURE 3.2: Schematic of a CI-QEM-protected imperfect CNOT gate equipped with a quantum channel precoder.

of gates applied both before and after the imperfect gate to be protected[4].

In order to obtain the inverse channel $\boldsymbol{M}$, we may consider a linear combination of predefined quantum operations, taking the following form

$$\boldsymbol{M} = \hat{\boldsymbol{C}}^{-1} = \sum_{l=1}^{L} \mu_l \boldsymbol{O}_l, \tag{3.28}$$

where $\boldsymbol{O}_l$ is the $l$-th quantum operation, while $\boldsymbol{\mu}_{\mathcal{C}} := [\mu_1 \ \dots \ \mu_L]^{\mathrm{T}}$ is the quasi-probability representation vector satisfying $\mathbf{1}^{\mathrm{T}}\boldsymbol{\mu}_{\mathcal{C}} = 1$. This linear combination may be rewritten as a probabilistic mixture of the quantum operations as follows:

$$\hat{\boldsymbol{C}}^{-1} = \|\boldsymbol{\alpha}\|_1 \sum_{l=1}^{L} s_l p_l \boldsymbol{O}_l, \tag{3.29}$$

where $s_l$ and $p_l$ are the $l$-th entries of $\boldsymbol{s}$ and $\boldsymbol{p}$, respectively, given by

$$\begin{aligned} p_i &= \frac{|\mu_i|}{\|\boldsymbol{\mu}_{\mathcal{C}}\|_1}, \\ \boldsymbol{s} &= \mathrm{sgn}\{\boldsymbol{\mu}\}. \end{aligned} \tag{3.30}$$

Note that the vector $\boldsymbol{p}$ describes a probability distribution.

The remaining task is to find the linear coefficients $\boldsymbol{\mu}_{\mathcal{C}}$. To this end, we may first construct a basis matrix $\boldsymbol{B}$, in which each column is the vectorized Pauli transfer matrix of a candidate circuit [38], namely, we have $[\boldsymbol{B}]_{:,l} = \mathrm{vec}(\boldsymbol{O}_l)$. Next, we determine the

---

[4] For the moment, we assume that the gates used to implement the quantum channel precoder are decoherence-free. The practical case of erroneous gates will be considered in Section 4.2.4.

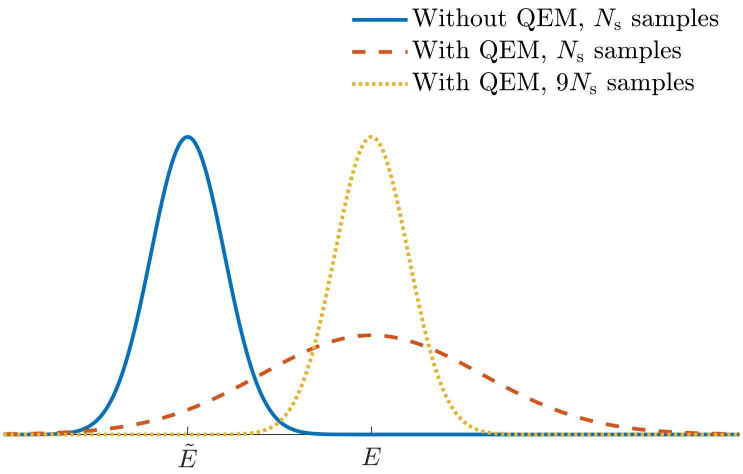


FIGURE 3.3: An illustration of the sampling overhead of CI-QEM. The curves represent the distribution of the computational results with/without CI-QEM.

coefficients as follows:

$$\boldsymbol{\mu}_{\mathcal{C}} = \boldsymbol{B}^{-1}\text{vec}\{\hat{\mathcal{C}}^{-1}\}. \tag{3.31}$$

Given the coefficients $\boldsymbol{\mu}_{\mathcal{C}}$, we can now express $\boldsymbol{M}$ as

$$\boldsymbol{M} = \sum_i [\boldsymbol{\mu}_{\mathcal{C}}]_i \text{vec}^{-1}\{[\boldsymbol{B}]_{:,i}\}. \tag{3.32}$$

This can be realized by applying the candidate circuit corresponding to $[\boldsymbol{B}]_{:,i}$ with probability $|[\boldsymbol{\mu}_{\mathcal{C}}]_i| \cdot \|\boldsymbol{\mu}_{\mathcal{C}}\|_1^{-1}$ and assigning a weight $\text{sgn}([\boldsymbol{\mu}_{\mathcal{C}}]_i) \cdot \|\boldsymbol{\mu}_{\mathcal{C}}\|_1$ to the measurement outcome. In light of this, $\boldsymbol{\mu}_{\mathcal{C}}$ is referred to as the quasi-probability representation [37] of channel $\mathcal{C}$.

#### 3.3.1.2 The Sampling Overhead of CI-QEM

In general, the probabilistic implementation of $\boldsymbol{M}$ will incur a sampling overhead. To elaborate, if we wish to compute the expectation value $J(\boldsymbol{\theta})$ to a certain accuracy, we have to operate the circuit a certain number of times (sampling from the output state vector). When the gates are perfect, the number of samples required is determined by the variance of the observable $\mathcal{H}$ given by

$$\text{Var}\{\mathcal{H}\} = \langle\psi(\boldsymbol{\theta})|\,\mathcal{H}^2\,|\psi(\boldsymbol{\theta})\rangle - (\langle\psi(\boldsymbol{\theta})|\,\mathcal{H}\,|\psi(\boldsymbol{\theta})\rangle)^2. \tag{3.33}$$

Upon assuming that the required accuracy is quantified in terms of its variance $\sigma^2$, this may be achieved using $N_s$ samples in the perfect gate scenario. After CI-QEM, the expected value remains unchanged. However, if the number of samples is kept fixed, CI-QEM will lead to a variance increase, since the entries in $\boldsymbol{\mu}_{\mathcal{C}}$ are not necessarily positive. Explicitly, the variance after CI-QEM is given by

$$\sigma^2_{\text{QEM}} = \|\boldsymbol{\mu}_{\mathcal{C}}\|_1^2\sigma^2, \tag{3.34}$$

where $\|\boldsymbol{\mu}_{\mathcal{C}}\|_1 \geq 1$. Therefore, in order to achieve the same accuracy, we have to sample every quantum gate $N_{\text{s}}(\|\boldsymbol{\mu}_{\mathcal{C}}\|_1^2 - 1)$ times additionally. To elaborate a little futher, we consider a toy example portrayed in Fig. 3.3. In this example, we assume that the error-free expectation value is $E$, and we assume that $\|\boldsymbol{\mu}_{\mathcal{C}}\|_1 = 3$. As can be observed from the figure, when the circuit is sampled $N_{\text{s}}$ times, the computational result without CI-QEM is randomly distributed around its mean value $\tilde{E}$, which deviates from the true value $E$. Having been corrected by CI-QEM, the mean value of the computational result equals to $E$, however, the variance of the result is enlarged by $\|\boldsymbol{\mu}_{\mathcal{C}}\|_1^2 = 9$ times. To ensure that the accuracy meets our requirement, we have to sample the circuit $N_{\text{s}}\|\boldsymbol{\mu}_{\mathcal{C}}\|_1^2 = 9N_{\text{s}}$ times, as illustrated by the dotted curve in Fig. 3.3. Empirical evidence has shown that applying quantum channel precoders is capable of reducing the sampling overhead for certain types of channels [38].

In the previous example, we have considered the case of a single channel $\mathcal{C}$. In general, a quantum circuit consists of $N_{\text{g}} > 1$ gates, which may be viewed as the cascade of $N_{\text{g}}$ channels, denoted by $\mathcal{C}_1, \ldots, \mathcal{C}_{N_{\text{g}}}$. To achieve the required computational accuracy, we have to sample the circuit for

$$\widetilde{N_{\text{s}}} = N_{\text{s}} \prod_{i=1}^{N_{\text{g}}} \|\boldsymbol{\mu}_{\mathcal{C}}\|_1^2 \tag{3.35}$$

times. To facilitate our analysis, we define the *sampling overhead* of a circuit as the additional number of samples imposed by CI-QEM, which equals to $\widetilde{N_{\text{s}}} - N_{\text{s}}$ in our previous example. It is noteworthy that the sampling overhead of a circuit grows exponentially with the number of gates.

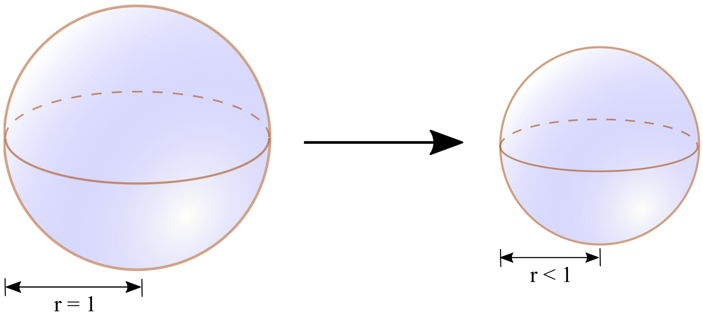
According to the previous discussions, we may use the following notion to characterize the sampling overhead incurred by a single channel $\mathcal{C}$ when compensated by CI-QEM.

**Definition 3.1** (Sampling Overhead Factor)**.** We define the sampling overhead factor (SOF) of a quantum channel $\mathcal{C}$ as

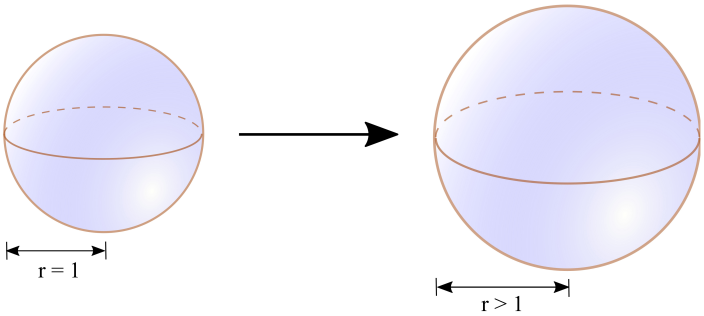
$$\gamma_{\mathcal{C}} \triangleq \|\boldsymbol{\mu}_{\mathcal{C}}\|_1^2 - 1. \tag{3.36}$$

*Remark* 3.2. When there is only a single gate with associated channel $\mathcal{C}$ in the circuit, from (3.35) we see that the sampling overhead of the circuit may be represented in terms of the sampling overhead factor (SOF) $\gamma_{\mathcal{C}}$ as $N_{\text{s}}\gamma_{\mathcal{C}}$. When there are more that one gates in the circuit, the sampling overhead of the circuit can be computed using the SOFs of the channels as follows:

$$\widetilde{N_{\text{s}}} - N_{\text{s}} = N_{\text{s}} \left[\prod_{i=1}^{N_{\text{g}}}(1 + \gamma_{\mathcal{C}_i}) - 1\right]. \tag{3.37}$$



(a) The depolarizing channel shrinks the Bloch sphere.



(b) The inverse of the depolarizing channel dilates the Bloch sphere.

FIGURE 3.4: Geometric illustration of a single-qubit depolarizing channel as well as the corresponding inverse channel.

To provide further intuitions about the SOF, let us consider the toy example of a single-qubit depolarizing channel $\mathcal{C}$, having the following operator-sum representation

$$\mathcal{C}(\rho) = 0.97\rho + 0.01\boldsymbol{S}_{\mathcal{X}}\rho\boldsymbol{S}_{\mathcal{X}} + 0.01\boldsymbol{S}_{\mathcal{Y}}\rho\boldsymbol{S}_{\mathcal{Y}} + 0.01\boldsymbol{S}_{\mathcal{Z}}\rho\boldsymbol{S}_{\mathcal{Z}},$$

which can be observed to have generalized gate error probability (GGEP) $\epsilon = 0.03$. The corresponding Pauli transfer matrix takes the following form

$$\boldsymbol{C} = \mathrm{diag}\left\{1, 0.96, 0.96, 0.96\right\},$$

and the inverse Pauli transfer matrix is given by

$$\boldsymbol{C}^{-1} = \mathrm{diag}\left\{1, 1.0417, 1.0417, 1.0417\right\}.$$

Geometrically, the channel $\mathcal{C}$ corresponds to a homogeneous shrinking of the Bloch sphere, making its radius 0.96 times the original radius, while the inverse channel $\mathcal{C}^{-1}$ corresponds to a homogeneous dilation extending the raduis to $1/0.96 = 1.0417$ times the original radius, as portrayed in Fig. 3.4.

To perform CI-QEM on the channel $\mathcal{C}$, we should first choose a basis. Note that the Pauli operators have the following Pauli transfer matrix representations

$$\begin{aligned} \boldsymbol{P}_{\mathcal{I}} &= \text{diag}\left\{1,1,1,1\right\}, \\ \boldsymbol{P}_{\mathcal{X}} &= \text{diag}\left\{1,1,-1,-1\right\}, \\ \boldsymbol{P}_{\mathcal{Y}} &= \text{diag}\left\{1,-1,1,-1\right\}, \\ \boldsymbol{P}_{\mathcal{Z}} &= \text{diag}\left\{1,-1,-1,1\right\}, \end{aligned} \tag{3.38}$$

which constitutes a complete basis of diagonal Pauli transfer matrices. In light of this, we may choose the Pauli operators as the basis. The corresponding quasi-probability representation can then be computed as

$$\boldsymbol{\mu}_{\mathcal{C}} = [1.03125 \ -0.01042 \ -0.01042 \ -0.01042]^{\mathrm{T}},$$

yielding $\|\boldsymbol{\mu}_{\mathcal{C}}\|_1 = 1.0625$. Therefore, given the basis we chose, the SOF of the channel $\mathcal{C}$ is given by

$$\gamma_{\mathcal{C}} = \|\boldsymbol{\mu}_{\mathcal{C}}\|_1^2 - 1 = 0.1289.$$

### 3.3.2 Symmetry-Based QEM

Broadly speaking, symmetry-based QEM methods may be viewed as low-complexity quantum error detection codes that have incomplete error detection capability. To elaborate, these methods are designed to detect the violations of certain symmetry conditions, either intrinsically embedded in the quantum algorithms themselves or artificially constructed. Once an error event is detected, one may simply discard the computational result, or project the result onto an "error-free subspace" using post-processing techniques, depending on the specific symmetry-based QEM methods used.

There are many symmetry conditions that can be exploited for QEM [36, 65, 127, 128]. State symmetry is a class of widely used symmetry conditions, which are typically represented using the stabilizer formalism, as follows:

$$\mathcal{S}\left|\psi\right\rangle = \left|\psi\right\rangle. \tag{3.39}$$

If we have a set of symmetry conditions for the state $|\psi\rangle$, it is clear from (3.39) that the corresponding stabilizers form a group $\mathscr{S}$, which is similar to the stabilizer group of quantum error correction codes. In this treatise, we term the symmetry conditions obeying the form of (3.39) as **state symmetries**, since they are characteristics of quantum states. As it will be discussed in Chapter 7, the concept of symmetry may also be generalized to **circuit symmetries**, which may offer additional error mitigation capability for practical quantum algorithms.

#### 3.3.2.1 Symmetry Verification

In order to exploit the state symmetries at a certain time instance during the computation, we should ensure that the state can be stabilized by the corresponding symmetry operators up until this moment. One of the possible approaches is to prepare an initial state satisfying all the symmetry conditions at the input of the circuit, as follows:

$$\mathcal{S}_i \left|\psi\right\rangle_{\text{in}} = \left|\psi\right\rangle_{\text{in}}, \; \forall \mathcal{S}_i \in \mathscr{S}. \tag{3.40}$$

Additionally, the gates constructing the circuit should be chosen for ensuring that the circuit commutes with all the symmetries, in the following sense

$$[\mathcal{S}_i, \mathcal{C}] = 0, \; \forall \mathcal{S}_i \in \mathscr{S}, \tag{3.41}$$

where $\mathcal{C}$ denotes the unitary operator representing the idealized noise-free quantum circuit. This ensures that

$$\mathcal{S}_i \left|\psi\right\rangle = \mathcal{S}_i \mathcal{C} \left|\psi\right\rangle_{\text{in}} = \mathcal{C} \mathcal{S}_i \left|\psi\right\rangle_{\text{in}} = \left|\psi\right\rangle, \tag{3.42}$$

hence the output state is also stabilized by the symmetries. In light of this, any error caused by noise or other imperfections of the circuit that violate the symmetry conditions may be identified by stabilizer checks and then mitigated.

As discussed in Section 3.2, identifying the specific form of the error would require from the stabilizer group to satisfy certain conditions, which typically implies a large number of stabilizer generators, especially when we aim for a high error correction capability. This would reduce the effective number of logical qubits, hence may not be computationally affordable in the context of NISQ computation. Thus we have to tolerate a limited error mitigation capability, in the sense that we might only detect the error event without recognizing its specific form. Consequently, the error can no longer be completely compensated for using unitary operations, which is in contrast to the recovery procedure of quantum error correction codes.

Mathematically speaking, given the stabilizer group $\mathscr{S}$, a natural technique of mitigating the potential error is to project the quantum state onto the "symmetry subspace" [129], using the following projection operator

$$\mathcal{T}_{\mathscr{S}} = \frac{1}{|\mathscr{S}|} \sum_{\mathcal{S} \in \mathscr{S}} \mathcal{S}. \tag{3.43}$$

Practically, this operator may be simplified by using the set $\mathscr{G}$ of generators of the stabilizer group $\mathscr{S}$

$$\mathcal{T}_{\mathscr{S}} = \frac{1}{|\mathscr{G}|} \sum_{\mathcal{S} \in \mathscr{G}} \mathcal{S}. \tag{3.44}$$

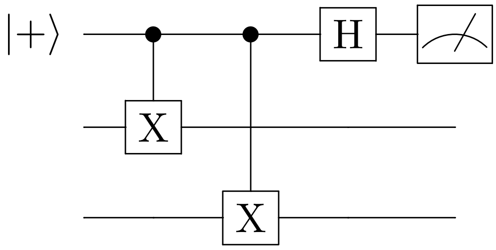


FIGURE 3.5: Illustration of "identify-discard" symmetry verification based on ancillas.

Apparently, we have $\mathcal{T}_{\mathscr{S}} |\psi\rangle = |\psi\rangle$ for legitimate states $|\psi\rangle$. The family of error mitigation methods based on these projection operators is termed as **symmetry verification** [36,127,128].

Similar to the syndrome measurement in quantum error correction codes, the most straightforward implementation of the projection operator is to use ancillas [36], as portrayed in Fig. 3.5. In this example, we verify a symmetry condition by measuring the corresponding $\boldsymbol{X}_1\boldsymbol{X}_2$ stabilizer using a single ancilla. Measuring a $|1\rangle$ on the ancilla implies violation of the symmetry condition, hence we may simply discard this circuit activation. It is also possible to dispense with using the ancilla, when the number of qubits is limited (termed as "in-line symmetry verification" in [36]), but at the cost of increased circuit depth. Assuming that the additional circuitry required for symmetry verification does not incur extra error, given an observable $\mathcal{O}$, the computational result of both ancilla-based and in-line symmetry verification may be expressed in a unified form as

$$r_{\mathrm{SV}} = \frac{\mathrm{Tr}\left\{\mathcal{O}\mathcal{T}_{\mathscr{S}}\rho\right\}}{\mathrm{Tr}\left\{\mathcal{T}_{\mathscr{S}}\rho\right\}}, \tag{3.45}$$

where $\rho$ is the density matrix representation of the state to be verified. For the simplicity of further discussion, we will refer to these two methods as "in-circuit" symmetry verification.

Similar to channel inversion based QEM, symmetry verification also suffers from a sampling overhead, due to the fact that the denominator $\mathrm{Tr}\left\{\mathcal{T}_{\mathscr{S}}\rho\right\}$ in (3.45) is typically less than 1 (otherwise the state $\rho$ is already in the symmetry subspace, and thus the symmetry verification has no further error mitigation effect on it). This increases the variance of the computational result, hence additional samples are required for ensuring that the output variance remains the same as that of the result without symmetry verification.

Alternatively, symmetry verification may also be implemented in a post-selection fashion, which would further reduce the computational error compared to that of in-circuit symmetry verification, since it does not need any addition circuitry (which may itself incur addition error). This may be achieved by treating the projection operator in (3.44) as a uniform distribution over the stabilizer generators of $\mathscr{S}$, and obtain the accuracy-improved computational result using Monte Carlo averaging over samples randomly drawn from this distribution [129].

#### 3.3.2.2 Virtual Distillation

In general, using symmetry verification requires some prior knowledge about the computational task in terms of its intrinsic symmetries. This is because constructing artificial symmetries is a non-trivial task, requiring appropriate basis transformations to the gates in the quantum circuit [36], similar to the problem of constructing logical gates in quantum error correction codes. Therefore, it would be more practical to rely solely on the intrinsic symmetries, but, this also imposes restrictions on the potential application scenarios of symmetry verification.

Permutation symmetry is an exception, which may be manipulated without introducing complex basis transformation circuits, hence has received much attention [65, 66, 123, 130, 131]. Specifically, if we could prepare $N$ identical states on different qubits, when there is no error, these states should remain the same upon rearranging the positions of the underlying qubits. Any error violating the permutation symmetry may then be identified under the framework of symmetry verification.

It has been shown in [65] that the stabilizer check circuits relying on the permutation symmetry may be implemented using controlled-swap gates (also known as quantum Fredkin gates [132]). The authors of [66] proposed essentially the same method, but with a particular focus on the scenario where only a pair of identical states are exploited. Additionally, they have proposed rather different circuit implementations, and termed the method as **virtual distillation**. Later versions of virtual distillation strike a flexible trade-off between the qubit overhead and the computational time in each circuit activation, by partially serializing the permutation symmetry verification procedure, which may be beneficial when the qubit resources are scarce [130, 131].

The trade-off between the error mitigation capability and the computational overhead of virtual distillation can be flexibly manipulated by appropriately adjusting the number of copies. Specifically, given an observable $\mathcal{O}$, in the same spirit as (3.45), the effect of virtual distillation relying on $N$ copies may be formulated as follows [66]

$$\frac{\mathrm{Tr}\left\{\mathcal{O}^{(1)}\mathcal{S}^{(N)}\rho^{\otimes N}\right\}}{\mathrm{Tr}\left\{\mathcal{S}^{(N)}\rho^{\otimes N}\right\}} = \frac{\mathrm{Tr}\left\{\mathcal{O}\rho^{N}\right\}}{\mathrm{Tr}\left\{\rho^{N}\right\}}, \tag{3.46}$$

where $\mathcal{O}^{(1)}$ denotes the observable $\mathcal{O}$ acting upon the first copy (but essentially an arbitrary copy) of the state $\rho$, and $\mathcal{S}^{(N)}$ denotes the cyclic shift operator applied to all

$N$ copies satisfying

$$\mathcal{S}^{(N)}\left(|\psi_1\rangle\otimes|\psi_2\rangle\dots|\psi_N\rangle\right)=|\psi_2\rangle\otimes|\psi_3\rangle\dots|\psi_1\rangle\,. \tag{3.47}$$

From the perspective of spectral analysis, when the quantum noise is not particularly strong, the ideal noise-free computational result should still be similar to the dominant eigenvector of $\rho$. Since the eigenvalues of $\rho$ are all non-negative and satisfy the normalization property and hence they sum up to 1, as the power $N$ tends to infinity, $\rho^N$ would tend to $|\psi_{\text{dom}}\rangle\langle\psi_{\text{dom}}|$, where $|\psi_{\text{dom}}\rangle$ denotes the dominant eigenvector of $\rho$, which is also the key observation exploited in the classical power method of matrix eigenvalue decomposition [133]. However, since the dominant eigenvector is not in perfect match with the noise-free result, there would be some error floor as $N\to\infty$, which becomes more significant when the main source of error in the circuit is coherent error instead of decoherence [123]. Fortunately, this error floor may be further mitigated by using virtual distillation in conjunction with other error mitigation techniques, such as the class of zero-noise extrapolation based methods, which will be discussed in Section 3.3.3.

### 3.3.3 Zero-Noise Extrapolation Based QEM

Reducing the physical noise level is a major challenge for all practical quantum computational systems. However, artificially increasing the noise level[5] may be relatively simple. Zero-noise extrapolation [37,38,74,134] is exactly such a class of QEM methods relying on artificial noise, which aims for extrapolating the computational results to the zero-noise limit using measurements obtained at different noise levels.

The key ingredients of zero-noise extrapolation are as follows:

1. **A perturbation-based approximate expansion of the computational error as a function of certain noise parameters**;
2. **Quantum control techniques associated with the noise parameters capable of manipulating their values**.

#### 3.3.3.1 Error Expansion Methods

The original form of zero-noise extrapolation proposed in the seminal contribution [37] relies on the Richardson extrapolation method of [135]. To be more specific, we may first represent the effect of noise as [135]:

$$\mathcal{C}=(1-\epsilon)\mathcal{I}+\epsilon\mathcal{N}, \tag{3.48}$$

---

[5] As will be discussed shortly, the "noise level" used in zero-noise extrapolation is not restricted to the probability of error or the power of the noise. In fact, it can be any parameter that determines the computational error.

where $\epsilon$ is the error probability, $\mathcal{I}$ denotes the identity operator, and $\mathcal{N}$ represents the undesired transformation acting on the quantum state when the error event occurs. Since the error rate $\epsilon$ is typically small (close to 0), given an observable $\mathcal{O}$, we may approximate its expectation value by a polynomial function of $\epsilon$ according to the Taylor expansion as follows:

$$\begin{aligned} r = \operatorname{Tr}\{\rho\mathcal{O}\} &:= F(\epsilon) \\ &= F(0) + \sum_{k=1}^{K} \alpha_k \epsilon^k + o(\epsilon^K), \end{aligned} \tag{3.49}$$

where $F(\epsilon)$ represents the noise-contaminated computational result at the noise level $\epsilon$, and $\alpha_k$, $k = 1 \dots K$ are unknown coefficients. Based on the previous assumptions, Richardson's extrapolation [135] may be used for approximating the noise free result $F(0)$ in terms of $K+1$ noisy measurement results, in the following form [135]

$$\begin{aligned} \hat{F}(0) &= \sum_{k=0}^{K} \mu_k F(\eta_k \epsilon_0) \\ &= F(0) + o(\epsilon^K), \end{aligned} \tag{3.50}$$

where $\epsilon_0$ is the minimum achievable noise level, $\eta_k$, $k = 0 \dots K$ is a list of multipliers adjusting the noise level satisfying $1 = \eta_0 < \eta_1 < \dots < \eta_K$, and the coefficients $\mu_k$, $k = 0 \dots K$ are determined by [135]

$$\mu_k = \prod_{j \neq k} \frac{\eta_j}{\eta_k - \eta_j}. \tag{3.51}$$

Similar to other QEM methods, Richardson's extrapolation also has a sampling overhead induced by the variance-increasing effect. In particular, its sampling overhead factor may be expressed as [74]

$$\gamma_{\mathrm{RE}} = \|\boldsymbol{\mu}\|_2^2 - 1, \tag{3.52}$$

where $\boldsymbol{\mu} := [\mu_0, \mu_1, \dots, \mu_K]^{\mathrm{T}}$.

When the number of gates in a quantum circuit is high, the polynomial approximation of Richardson's extrapolation becomes less accurate. This motivates the proposal of exponential extrapolation, which treats the dependence between the noisy computational result and the noise level as an exponential function [38, 136]. The plausibility of exponential extrapolation may be understood by considering a circuit consisting of $N_{\mathrm{G}}$ gates, each associated with a channel given by (3.48). The effect of this circuit may then be formulated as [136]

$$\prod_{n=1}^{N_{\mathrm{G}}} \mathcal{C}_n \mathcal{U}_n = \sum_{k=0}^{N_{\mathrm{G}}} (1-\epsilon)^{N_{\mathrm{G}}-k} \epsilon^k \sum_i \mathcal{E}_k^{(i)}, \tag{3.53}$$

where $\mathcal{E}_k^{(i)}$ represents one of the error patterns (i.e., the $i$-th error pattern) containing $k$ errors. Upon introducing the notations of $p_k := (1-\epsilon)^{N_\mathrm{G}-k}\epsilon^k$ and $\mathcal{E}_k := \binom{N_\mathrm{G}}{k}^{-1}\sum_i \mathcal{E}_k^{(i)}$, we may further rewrite (3.53) as

$$\prod_{n=1}^{N_\mathrm{G}} \mathcal{C}_n\mathcal{U}_n = \sum_{k=0}^{N_\mathrm{G}} p_k\mathcal{E}_k. \tag{3.54}$$

Note that $p_k$ may be viewed as the occurrence probability of $k$ errors in a binomial distribution, which may be approximated by a Poisson distribution given by

$$p_k \approx \frac{\epsilon N_\mathrm{G}}{k!}\exp(-\epsilon N_\mathrm{G}), \tag{3.55}$$

when the number of gates $N_\mathrm{G}$ is large. Hence the effect of the circuit in (3.53) may then be rearranged as

$$\prod_{n=1}^{N_\mathrm{G}} \mathcal{C}_n\mathcal{U}_n = \exp(-\epsilon N_\mathrm{G})\sum_{k=0}^{N_\mathrm{G}} \frac{\epsilon N_\mathrm{G}}{k!}\mathcal{E}_k, \tag{3.56}$$

where the factor $\exp(-\epsilon N_\mathrm{G})$ decreases exponentially with the noise level $\epsilon$.

#### 3.3.3.2 Noise Level Manipulation

Note that in Section 3.3.3.1, the noise level $\epsilon$ is a rather abstract parameter, as we have not assumed any physical interpretation of it. In the seminal contribution [37], the noise level is assumed to be a scaling factor of the Hamiltonian that generates the quantum circuit. In general, the evolution of a quantum state according to a quantum circuit may be expressed in terms of the master equation as [37]

$$\frac{\partial}{\partial t}\rho_\lambda(t) = -j[\mathcal{H}(t), \rho_\epsilon(t)] + \lambda\mathcal{L}\{\rho_\lambda(t)\}, \tag{3.57}$$

where $\rho_\epsilon(t)$ represents the quantum state at time $t$, $\mathcal{H}(t)$ denotes the Hamiltonian at time $t$, $\mathcal{L}(\cdot)$ is the Lindblad operator [137] modelling the noise effect, and $\lambda$ denotes the (not yet manipulated) strength of noise. If we prolong the evolution time to be $m$ times that of the original time, and reduce the evolution strength to be $1/m$ of the original strength, the Hamiltonian becomes $1/m\mathcal{H}(t/m)$, leading to the state $\rho_{m\lambda}(t)$. This implies that the noise level is increased from $\lambda$ to $m\lambda$. This procedure is typically implemented by reshaping the controlling pulse of quantum gates in experiments [37, 134, 138].

The Hamiltonian-based method of [37] offers an accurate control of the noise parameter, but it requires pulse-level control of the quantum circuits, which may not be accessible for the end-users of quantum computers. To this end, gate-level methods have also been proposed [139]. As an intuitive example, if we wish to increase the error rate of gate $\mathcal{U}$ satisfying $\mathcal{U}^m = \mathcal{I}$ (e.g., the Pauli-X operator satisfies $\mathcal{X}^2 = \mathcal{I}$), we may simply repeat the gate $(km+1)$ times, where $k$ is an integer. The drawback of this method is that it can only boost the error component that commutes with the repeated gate, while the non-commuting component is left unchanged. Consequently, zero-noise extrapolation based on gate repetition can only mitigate the commuting errors. Another issue is that it only offers discrete values of the noise level, which can be partially resolved by treating $k$ as a random variable [140, 141].

### 3.3.4 Learning-Based QEM

Most of the aforementioned QEM techniques either require prior knowledge about the noise (e.g., channel inversion and symmetry verification), or assume that the noise satisfies certain properties (e.g., zero-noise extrapolation). Since both noise sensing and error mitigation would incur sampling overhead, an integration of both tasks may reduce the overhead. As machine learning and data-driven modelling becoming increasingly successful in diverse areas [142–145], they have also been applied to QEM for extracting information about the noise characteristics from a set of training data before the main computational task is performed.

Considering the practicality of learning-based QEM, the training data should be obtained at a negligible cost compared to that of the main computational task. In light of this, the Clifford gates (i.e., quantum operations forming the Clifford group [146]) play a significant role in learning-based QEM, due to the celebrated Gottesman–Knill theorem [147], formally stated as follows:

**Theorem 3.3** (Gottesman–Knill)**.** *A quantum circuit satisfying the following conditions:*

1. *Initial states are the eigenstates of Pauli operators;*
2. *They only have Clifford gates;*
3. *They rely on measurements in the computational basis,*

*can be simulated in polynomial time on classical computers.*

*Proof.* Please refer to [147]. □

For the simplicity of further discussion, we will refer to quantum circuits satisfying the conditions in the Gottesman–Knill theorem as “Clifford circuits”. Given a specific Clifford circuit and an observable, we may generate two sets of data: a set of noisy expectation values of the observable obtained using the quantum computer, denoted by $x_{\text{noisy}}$, and an associated set of ideal error-free expectation values obtained using efficient classical simulation algorithms, denoted by $x_{\text{ideal}}$. Machine learning methods may then be applied to fit a model that describes the mapping from $x_{\text{noisy}}$ to $x_{\text{ideal}}$,

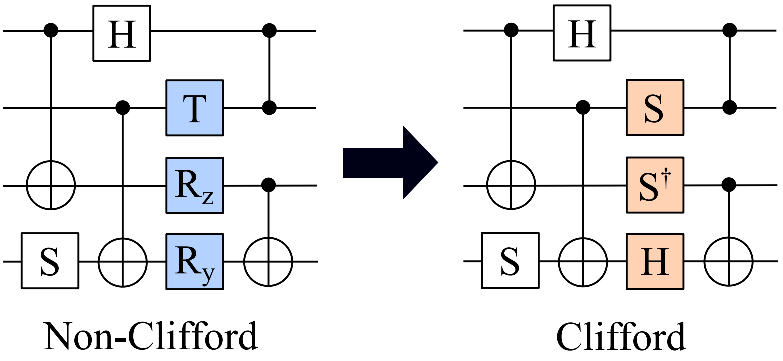


FIGURE 3.6: Illustration of a computational circuit (which is non-Clifford) and the Clifford circuit obtained by replacing the non-Clifford gates by corresponding best-fit Clifford gates. In this example, we assume that the $\mathcal{R}_z$ gate represents a $-60$-degree rotation around the Z-axis of the Bloch sphere, while the $\mathcal{R}_y$ gate represents a 60-degree rotation around the Y-axis.

denoted by $\boldsymbol{x}_{\text{ideal}} = f(\boldsymbol{x}_{\text{noisy}}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is some model parameter. Hence the effect of noise on the computational result is mitigated. Formally, the above-mentioned model may be learnt by minimizing a cost function taking the following form [64]

$$J(\boldsymbol{\theta}) = \sum_{n} \left[ f(\boldsymbol{x}_{\text{noisy}}, \boldsymbol{\theta}) - \boldsymbol{x}_{\text{ideal}} \right]^2 . \tag{3.58}$$

This is the basic idea of the so-called Clifford data regression proposed in [64], whose performance has been experimentally quantified based on the IBM quantum computer.

The main practical issue concerning Clifford data regression is that the mapping between the noisy and the ideal data would be different from $f(\boldsymbol{x}_{\text{noisy}}, \boldsymbol{\theta})$ in practice, since the quantum circuit used in actual computation is typically not a Clifford circuit (otherwise there will not be any computational advantage compared to classical algorithms, according to the Gottesman–Knill theorem). The authors of [64] have suggested a potential solution to this issue, which essentially fits the Clifford circuit to the actual computational circuit. Specifically, we may construct the Clifford circuit by replacing the non-Clifford gates in the computational circuit by Clifford circuits that are close in terms of certain distance metrics to the original gates, as portrayed in Fig. 3.6. To elaborate, we see that the $\mathcal{T}$ gate representing a 45-degree rotation around the Z-axis of the Bloch sphere is replaced by the $\mathcal{S}$ gate, which represents a 90-degree rotation around the Z-axis. Similarly, the $\mathcal{R}_z$ gate representing a $-60$-degree rotation around the Z-axis is replaced by the $\mathcal{S}^\dagger$ gate, which represents a $-90$-degree rotation around the Z-axis. Finally, the $\mathcal{R}_y$ gate representing a 60-degree rotation around the Y-axis, hence is approximated by the Hadamard gate representing a 90-degree rotation around the Y-axis. These replacement gates are chosen from the set of Clifford gates by minimizing the so-called "diamond distance" to the original gates [64]. Specifically, the diamond distance between two quantum operations $\mathcal{A}$ and $\mathcal{B}$ is defined as [48]

$$\|\mathcal{A}-\mathcal{B}\|_{\diamondsuit}=\max_{\rho}\|(\mathcal{A}\otimes\boldsymbol{I})\rho-(\mathcal{B}\otimes\boldsymbol{I})\rho\|_1. \tag{3.59}$$

The idea of training the refinement model $f(\boldsymbol{x}_{\text{noisy}},\boldsymbol{\theta})$ on the set of Clifford gates has also been generalized in conjunction with channel inversion based QEM [120]. Similar to the original channel inversion based QEM, the method proposed in [120] inserts some additional gates after each gate in the original computational circuit. The additional gates are specifically chosen for minimizing a cost function similar to (3.58), where the mapping $f(\boldsymbol{x}_{\text{noisy}},\boldsymbol{\theta})$ is implemented by these additional gates. In order to simplify the learning process, one may apply Pauli twirling to each gate in the computational circuit, which effectively turns the quantum channels associated with these gates to Pauli channels. Consequently, the effective number of parameters to be estimated is reduced, since in general a Pauli channel over $n$ qubits has $4^n-1$ free parameters [1], while the number of free parameters of a general quantum channel would be on the order of $O(16^n)$ [1].

## 3.4 Algorithm-Level Quantum Error Mitigation

As discussed in Section 3.1.2, one of the main algorithm-level source of error is the approximation error caused by the non-commutativity between Hamiltonian components in Hamiltonian simulations. For example, the first-order Trotter approximation given by [75]

$$e^{j\tau\mathcal{H}}=\left(\prod_{k=1}^{K}e^{j\tau/N_\tau\mathcal{H}_k}\right)^{N_\tau}+O(N_\tau^{-1}), \tag{3.60}$$

has an approximation error on the order of $O(N_\tau^{-1})$. To mitigate this error, the authors of [63] provided the key observation that upon denoting $\epsilon=1/N_\tau$, (3.60) may be rewritten as

$$e^{j\tau\mathcal{H}}=\left(\prod_{k=1}^{K}e^{j\tau\epsilon\mathcal{H}_k}\right)^{1/\epsilon}+O(\epsilon). \tag{3.61}$$

Observe that $\epsilon$ may be viewed as the level of the error. This suggests that the zero-noise extrapolation method introduced in Section 3.3.3 may be applied to extrapolate the approximated computational result $\left(\prod_{k=1}^{K}e^{j\tau\epsilon\mathcal{H}_k}\right)^{1/\epsilon}$ towards the $\epsilon\rightarrow 0$ limit. Specifically, given an observable $\mathcal{O}$, we may consider the expansion

$$\begin{aligned}\operatorname{Tr}\{\rho\mathcal{O}\}:&=F(\epsilon)\\&=F(0)+\sum_{k=1}^{K}\alpha_k\epsilon^k+o(\epsilon^K).\end{aligned} \tag{3.62}$$

In light of this, one may then use Richardson's extrapolation method to obtain an estimate of the error-free result $F(0)$, based on multiple measurement outcomes corresponding to different $\epsilon$, using the following inverse expansion [63]:

$$\begin{aligned}\hat{F}(0) &= \sum_{k=0}^{K} \mu_k F(\eta_k \epsilon_0) \\ &= F(0) + o(\epsilon^K).\end{aligned} \tag{3.63}$$

The manipulation of $\epsilon$ is straightforward, since it is simply the reciprocal of the number of Trotter steps $N_\tau$.

Another method mitigating the algorithmic error of Hamiltonian simulation is the quantum circuit Monte Carlo [148], which is based on a "summation over unitaries" decomposition of the time evolution operator $\exp(j\tau/N_\tau \mathcal{H})$ at a single time step $\tau/N_\tau$, as follows

$$e^{j\tau/N_\tau \mathcal{H}} = \sum_s c_s \mathcal{U}_s, \tag{3.64}$$

where $c_s$ are coefficients, and $\mathcal{U}_s$ are unitary operators. Note that there is a large degree of freedom in choosing the specific form of the decomposition (3.64). A straightforward choice is the Pauli decomposition given by

$$e^{j\tau/N_\tau \mathcal{H}} = \frac{1}{2^n} \sum_{\mathbf{S}_i^{(n)} \in \mathscr{P}^n} \mathrm{Tr}\left\{ \mathbf{S}_i^{(n)} e^{j\tau/N_\tau \mathcal{H}} \right\} \mathbf{S}_i^{(n)}, \tag{3.65}$$

where $\mathscr{P}^n$ denotes the Pauli group over $n$ qubits. However, this is typically computationally inefficient, since the complexity of computing all the coefficients $\mathrm{Tr}\left\{ \mathbf{S}_i^{(n)} e^{j\tau/N_\tau \mathcal{H}} \right\}$ would be prohibitive.

To construct a computationally efficient decomposition of $\exp(j\tau/N_\tau \mathcal{H})$, the authors of [148] conceived a modification of the Trotterization method, termed as "Pauli operator expansion", which is based on the following decomposition

$$e^{j\tau/N_\tau \mathcal{H}} = \mathcal{T}_{\mathrm{L}} \mathcal{V} \mathcal{T}_{\mathrm{R}}, \tag{3.66}$$

where $\mathcal{T}_{\mathrm{L}}$ and $\mathcal{T}_{\mathrm{R}}$ are Trotterization operators in a similar product form as (3.60), while $\mathcal{V}$ is a correction term. The Pauli operator expansion relies on a Taylor expansion of the correction term $\mathcal{V}$ in the form of

$$\mathcal{V} = \mathcal{I} + j\mathcal{L} + \mathcal{W}, \tag{3.67}$$

where $\mathcal{I}$ is the identity operator, $\mathcal{L}$ is the first-order term, and $\mathcal{W}$ denotes the higher order term that will be neglected in the circuit implementation.

Geometrically, multiplying a quantum state by a certain operator may be viewed as

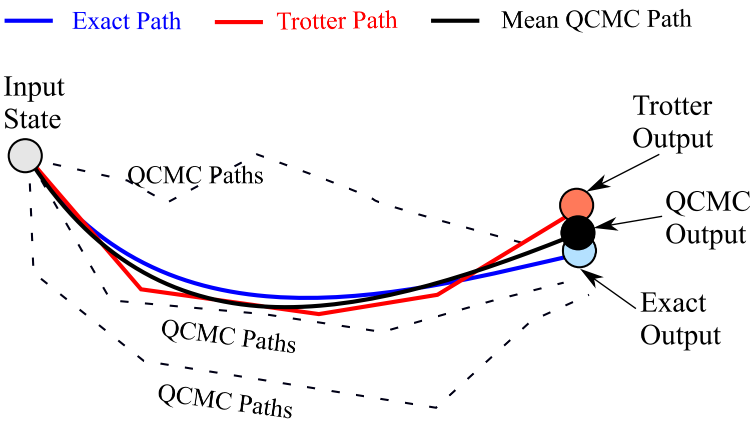


FIGURE 3.7: Geometric interpretation of the Trotter approximation and the quantum circuit Monte Carlo (QCMC) method.

"taking a step" in the state space, while a summation of operators may be viewed as averaging the output states generated by multiple paths. Hence, as portrayed in Fig. 3.7, the Trotter approximation in (3.60) may be viewed as approximating the exact state evolution using a single path, since it is merely constituted by multiplications. By contrast, the Pauli operator expansion in (3.65) may be viewed as a weighted summation over multiple paths (i.e., the "path integral" [149]), where the weights are determined by the correction term $\mathcal{V}$. The parlance "quantum circuit Monte Carlo" arises from the fact that this weighted summation may be implemented by sampling the quantum circuits corresponding to the different evolution paths from a probabilistic distribution determined by the weights, and taking the average of the outcomes [148].

To elaborate further, let us consider the second-order Pauli operator expansion having the following form

$$e^{j\tau/N_\tau \mathcal{H}} = \mathcal{T}_1\left(-\frac{\tau}{2N_\tau}\right)^\dagger \mathcal{V}_2(\tau/N_\tau)\mathcal{T}_1\left(\frac{\tau}{2N_\tau}\right), \tag{3.68}$$

where $\mathcal{S}_1(t)$ is the Trotterization operator defined as $\mathcal{S}_1(t) = \prod_{k=1}^{K} e^{jt\mathcal{H}_k}$, while the correction term $\mathcal{V}_2(\tau/N_\tau)$ is given by

$$\mathcal{V}_2(t) = \mathcal{I} + \underbrace{\left(\mathcal{F}_2^{(3)}(t) + \mathcal{F}_2^{(5)}(t)\right)}_{\text{First−order terms}} + \mathcal{W}_2(t), \tag{3.69}$$

with $\mathcal{F}_2^{(k)}$ having a complex form given as (39) in [148]. Comparing the second-order Trotterization (3.8) with the second-order Pauli operator expansion (3.68), we see that their only difference resides in the correction term $\mathcal{V}_2(\tau/N_\tau)$. However, the second-order Pauli operator expansion has an algorithmic error on the order of $O(N_\tau^{-6})$, which is substantially lower than the $O(N_\tau^{-2})$ error scaling of the second-order Trotterization.

This implies that the correction term $\mathcal{V}_2(\tau/N_\tau)$ serves as an efficient algorithm-level error mitigator.

Besides the Trotterization error, another common source of algorithm-level error is the probabilisitc sampling error due to the probabilistic nature of quantum algorithms, as have been discussed in Section 3.1.2. Intuitively, quantum algorithms correspond to unitary operators, which may be viewed as rotations in the space of quantum states. Sampling errors would thus occur when the rotations do not finish at computational basis states. This is common when the depth of the circuit is unknown before the computation, as seen in quantum search algorithms [13–15], or the unknown parameter is not an integer multiple of the number of computational basis states, as seen in quantum phase estimation [91, 150, 151]. For quantum search algorithms, this issue is typically addressed by executing the same algorithm for multiple times, but gradually increasing the search range after each execution [14]. For quantum phase estimation, the algorithm-level error may be interpreted as a quantum version of the “spectral leakage” problem in classical signal processing, which will be discussed in more detail in Chapter 8.

# Chapter 4

# Channel-Inversion Based QEM: Sampling Overhead Analysis

As discussed in Section 3.3.1.2, applying CI-QEM would incur a computational overhead, namely the sampling overhead. It originates from the fact that the "inverse channel" is typically not CPTP, unless the original channel is of unitary nature, and hence it does not impose decoherence [37, 38]. Consequently, CI-QEM leads to an increased variance in the final computational result, hence additional measurements are required at the output quantum state for achieving a satisfactory accuracy. In this chapter, we provide a comprehensive analysis of the sampling overhead of CI-QEM.

From a broader perspective, the sampling overhead has a profound relationship with the computational complexity theory. In effect, increasing the number of measurements will slow down the computation process. As the depth of the quantum circuit grows, the sampling overhead may accumulate dramatically. Ultimately, the benefit of quantum speedup will be neutralized for computation tasks that require extremely long coherence time. In its essence, CI-QEM may be viewed as an emulation of error-free quantum circuits with the aid of realistic quantum circuits degraded by decoherence. Since decoherence is widely recognized to perturb the quantum states (i.e., would ultimately turn quantum states into classical states), the sampling overhead may be viewed as a complexity penalty, when emulating quantum computers using "semi-classical" devices corresponding to the noisy quantum circuits.

In general, the sampling overhead required depends on the channel characteristics. Naturally, a fundamental question concerning the practicality of CI-QEM is: "Can we predict and control the sampling overhead given a limited number of channel parameters?" In this chapter, we investigate this deep-rooted research question from both theoretical and practical perspectives. We first introduce the notion of the so-called SOF for characterizing the sampling overhead incurred by a quantum channel, and then provide a comprehensive analysis of the SOF of general CPTP channels. Finally, we discuss potential techniques of reducing the SOF of quantum channels.

The rest of this chapter is organized as follows. In Section 4.1, we define some quality metrics of quantum channels, which will be applied extensively in the subsequent sampling overhead analysis. Based on this formulation, we analyse the SOF of uncoded quantum gates in Section 4.2. In Section 4.3, we conceive and analyse the amalgam of CI-QEM and QECCs as well as QEDCs, which will be referred to as the QECC-CI-QEM and QEDC-CI-QEM schemes, respectively. The accuracy of the analytical results is then verified by numerical simulations in Section 4.4. Finally, we conclude in Section 4.5.

## 4.1 Quality Metrics of Quantum Channels

A key quality indicator of a quantum channel is its average fidelity. As the terminology "average fidelity" suggests, it is the fidelity between the input and the output states, integrated over the space of all legitimate input states. Formally, the average fidelity of a quantum channel $\mathcal{C}$ is defined as [152]

$$\bar{F}(\mathcal{C}) = \int \langle\psi|\, \mathcal{C}(|\psi\rangle\langle\psi|)\, |\psi\rangle \ \mathrm{d}|\psi\rangle\,, \tag{4.1}$$

where the integral is taken over the Haar measure on the state space. This integral formula can be substantially simplified, when we have an orthonormal basis spanning the state space. Specifically, assume that the unitary operators $\{\mathcal{U}_j/\sqrt{d}\}$ (where $d$ denotes the dimensionality of the state space) constitute an orthonormal basis of the state space, it has been shown in [152] that the average fidelity can be expressed as

$$\bar{F}(\mathcal{C}) = \frac{\sum_j \mathrm{Tr}\left\{\boldsymbol{U}_j^\dagger \mathcal{C}(\boldsymbol{U}_j)\right\} + d^2}{d^2(d+1)}. \tag{4.2}$$

Using the Pauli transfer matrix representation, we have the following closed-form expression for the average fidelity of $\mathcal{C}$.

**Proposition 4.1.** *Given the Pauli transfer matrix $\boldsymbol{C}$ of the quantum channel $\mathcal{C}$, the average fidelity of $\mathcal{C}$ can be written in closed form as*

$$\bar{F}(\mathcal{C}) = \frac{\mathrm{Tr}\{\boldsymbol{C}\} + 2^n}{4^n + 2^n}, \tag{4.3}$$

*where $n$ is the number of qubits that $\mathcal{C}$ acts upon.*

*Proof.* Under the Pauli transfer matrix representation, a natural choice for the unitary operators $\boldsymbol{U}_j$ in (4.2) is constituted by the Pauli operators, namely $\boldsymbol{U}_j = \boldsymbol{S}_j$. Thus (4.2) may now be rewritten as

$$\bar{F}(\mathcal{C}) = \frac{\sum_{j=1}^{4^n} \mathrm{Tr}\left\{\boldsymbol{S}_j\mathcal{C}(\boldsymbol{S}_j)\right\} + 4^n}{4^n(2^n+1)}, \tag{4.4}$$

where we used the fact that $\boldsymbol{S}_i^\dagger = \boldsymbol{S}_i$ holds for Pauli operators $\boldsymbol{S}_i$. According to the definition of the PTM (2.17), we have

$$\mathrm{Tr}\left\{\boldsymbol{C}\right\} = \frac{1}{2^n}\sum_{j=1}^{4^n} \mathrm{Tr}\left\{\boldsymbol{S}_j\mathcal{C}(\boldsymbol{S}_j)\right\}, \tag{4.5}$$

and hence

$$\begin{aligned}\bar{F}(\mathcal{C}) &= \frac{2^n\mathrm{Tr}\left\{\boldsymbol{C}\right\} + 4^n}{2^n(4^n+2^n)} \\ &= \frac{\mathrm{Tr}\{\boldsymbol{C}\} + 2^n}{4^n+2^n},\end{aligned} \tag{4.6}$$

which completes the proof. □

In general, $\bar{F}(\mathcal{C})$ satisfies $0 \le \bar{F}(\mathcal{C}) \le 1$, and $1 - \bar{F}(\mathcal{C})$ is often referred to as the "average infidelity" of $\mathcal{C}$ [152].

For Pauli channels, another important quality metric is the gate error probability (GEP), namely the probability that the output state does not coincide with the input state. For example, for the following channel

$$\mathcal{C}(\rho) = (1-p)\rho + \frac{p}{3}\boldsymbol{S}_{\mathcal{X}}\rho\boldsymbol{S}_{\mathcal{X}} + \frac{p}{3}\boldsymbol{S}_{\mathcal{Y}}\rho\boldsymbol{S}_{\mathcal{Y}} + \frac{p}{3}\boldsymbol{S}_{\mathcal{Z}}\rho\boldsymbol{S}_{\mathcal{Z}}, \tag{4.7}$$

the GEP is $p$. Many important results on quantum coding, including the threshold theorem, are based on GEP.

A somewhat perplexing issue is that the GEP is inconsistent with the average fidelity. More precisely, for a Pauli channel $\mathcal{C}$, we have $\bar{F}(\mathcal{C}) \neq 1 - \mathrm{GEP}$. To avoid the difficulty of using two different metrics for Pauli and non-Pauli channels, in this chapter, we introduce a generalization of the GEP, which will be referred to as the GGEP hereafter. Specifically, we define the GGEP of channel $\mathcal{C}$ as

$$\epsilon(\mathcal{C}) = 1 - \frac{1}{4^n}\mathrm{Tr}\{\boldsymbol{C}\}. \tag{4.8}$$

As a channel quality metric, GGEP has the following advantages.

1. When $\mathcal{C}$ is a Pauli channel, the GGEP degenerates to the conventional GEP, which is $p$ in (4.7).

2. For a general channel $\mathcal{C}$, which is not necessarily a Pauli channel, the GGEP is proportional to the average infidelity of $\mathcal{C}$, in the sense that

$$\epsilon(\mathcal{C}) = \left(1 + \frac{2^n}{4^n}\right)[1 - \bar{F}(\mathcal{C})]. \tag{4.9}$$

   Thus an operation preserving the average fidelity would also preserve the GGEP.

## 4.2 Sampling Overhead Factor Analysis for Uncoded Quantum Gates

In this section, we investigate the SOF of quantum gates that are not protected by quantum codes. This will also lay the foundation for the analysis of coded quantum gates in Section 4.3.

The SOF, in essence, may also be viewed as a specific characteristic of the representation of a quantum channel under a specific basis, as implied by (3.31) and (3.36). Hence, the particular choice of basis will certainly have an impact on it. Considering realistic restrictions and aiming for simplifying our analysis, we make the following assumptions concerning the choices of basis.

1. The basis vectors should correspond to legitimate quantum operations in order to be implementable. Formally, we assume that the basis vectors are the vectorized Pauli transfer matrices of completely positive trace-nonincreasing (CPTnI) operators, for which the operation components of (2.12) satisfy

$$\sum_i \boldsymbol{K}_i^\dagger \boldsymbol{K}_i \preccurlyeq \boldsymbol{I}. \tag{4.10}$$

   This includes perfect gates (unitary operators), imperfect gates (CPTP operators), and measurements (trace-decreasing operators).
2. We assume that the basis always includes all vectorized Pauli operators. This is not very restrictive for most existing quantum computers, since Pauli gates are one of their most fundamental building blocks.

The other (potentially more important) factor influencing the SOF is the quantum channel itself. Various channel models have been proposed in the literature, such as depolarizing channels, phase damping channels, amplitude damping channels, etc. [38,44,153]. To maintain the generality of our treatment, we do not explicitly consider a specific channel model, but rather a general CPTP channel.

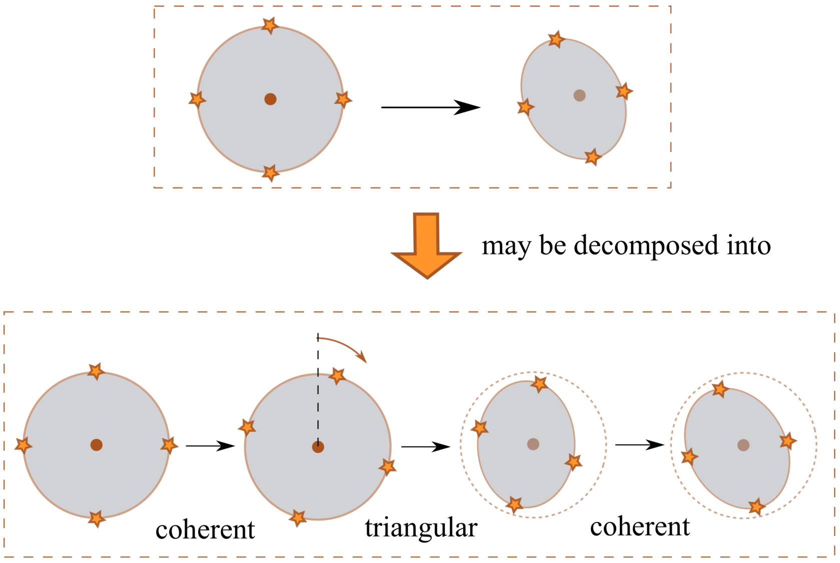


FIGURE 4.1: Geometric interpretation of the coherent–triangular decomposition of a signle-qubit CPTP channel on the Bloch sphere, detailed in Lemma 4.2. For better illustration, we only plot a cross-section of the Bloch sphere.

### 4.2.1 Coherent–Triangular Decomposition of Memoryless CPTP channels

Without loss of generality, we assume that

*Assumption* 1. The first row and the first column in a Pauli transfer matrix corresponds to the identity operator $\mathcal{I}^{\otimes n}$ in the $n$-qubit Pauli group.

Note that for a valid density matrix $\rho$, the condition $\mathrm{Tr}\{\rho\} = 1$ is always satisfied. According to Assumption 1, this implies that for the corresponding vector representation $\boldsymbol{x}$, we have

$$\boldsymbol{x} = \frac{1}{\sqrt{2^n}}[1,\ \tilde{\boldsymbol{x}}^{\mathrm{T}}]^{\mathrm{T}}, \tag{4.11}$$

where $\tilde{\boldsymbol{x}} \in \mathbb{R}^{4^n-1}$, according to (2.18). The dimensionality of $\boldsymbol{x}$ is $4^n$, because the number of Pauli operators over $n$ qubits is $4^n$. Thus for any trace-preserving channel, we have

$$\boldsymbol{C} = \begin{bmatrix} 1 & \boldsymbol{0}^{\mathrm{T}} \\ \boldsymbol{b} & \tilde{\boldsymbol{C}} \end{bmatrix}, \tag{4.12}$$

which amounts to the following result

$$\boldsymbol{C}\boldsymbol{x} = \frac{1}{\sqrt{2^n}}[1,\ \tilde{\boldsymbol{C}}\tilde{\boldsymbol{x}} + \boldsymbol{b}]^{\mathrm{T}}. \tag{4.13}$$

It can now be seen from (4.11), (4.12) and (4.13) that a CPTP channel can be viewed as an *affine transformation* in the $(4^n - 1)$-dimensional space spanned by the Pauli operators excluding the identity. In this regard, we have the following result for single-qubit channels, whose geometric interpretation is demonstrated in Fig. 4.1.

**Lemma 4.2.** *Any single-qubit CPTP channel can be expressed as the composition of (up to) two coherent channels*[1] *and a triangular channel, meaning that*

$$\begin{aligned} \boldsymbol{C} &= \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\mathrm{T}} \\ &= \begin{bmatrix} 1 & \boldsymbol{0}^{\mathrm{T}} \\ \boldsymbol{0} & \tilde{\boldsymbol{U}} \end{bmatrix} \begin{bmatrix} 1 & \boldsymbol{0}^{\mathrm{T}} \\ \tilde{\boldsymbol{U}}^{\mathrm{T}}\boldsymbol{b} & \tilde{\boldsymbol{D}} \end{bmatrix} \begin{bmatrix} 1 & \boldsymbol{0}^{\mathrm{T}} \\ \boldsymbol{0} & \tilde{\boldsymbol{V}} \end{bmatrix}^{\mathrm{T}}, \end{aligned} \tag{4.14}$$

*where $\tilde{\boldsymbol{U}}$ and $\tilde{\boldsymbol{V}}$ are a pair of unitary matrices, and $\tilde{\boldsymbol{D}}$ is a diagonal matrix. The matrix $\boldsymbol{D}$ corresponds to the triangular channel, while the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ represent the coherent channels.*

*Proof.* Consider the singular value decomposition of $\tilde{\boldsymbol{C}}$ given by

$$\tilde{\boldsymbol{C}} = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{D}}\tilde{\boldsymbol{V}}^{\mathrm{T}}. \tag{4.15}$$

From (4.12) we obtain directly that $\boldsymbol{D}$ is a triangular channel, and hence it now suffices to show that both $\boldsymbol{U}$ and $\boldsymbol{V}$ can be implemented by unitary gates. Since the entries of Pauli transfer matrices are all real numbers [107], the matrices $\tilde{\boldsymbol{U}}$ and $\tilde{\boldsymbol{V}}$ are both $3 \times 3$ orthogonal matrices, corresponding to the three-dimensional rotations around the Bloch sphere belonging to the special orthogonal group SO(3). They can be implemented by single-qubit unitary gates belonging to SU(2) due to the SO(3)-SU(2) homomorphism. □

Compared to the triangular component, the coherent component of a CPTP channel might be easier to deal with, since their effect may be compensated by using unitary gates. This implies that if the unitary gates designed for the compensation are error-free, the effect of coherent channels may be reversed without any sampling overhead. By contrast, the triangular component may have to be compensated by using probabilistic gates, hence imposes overhead.

It is known that Lemma 4.2 does not hold for general multi-qubit channels [154]. Nevertheless, it is applicable to the case where the channel $\boldsymbol{C}$ is memoryless, hence it can be described by the tensor product of single-qubit channels. To see this, we may rewrite a memoryless channel as $\boldsymbol{C} = \boldsymbol{C}_1 \otimes \boldsymbol{C}_2 \otimes \ldots \otimes \boldsymbol{C}_n$, and for each $\boldsymbol{C}_i$ we have $\boldsymbol{C}_i = \boldsymbol{U}_i\boldsymbol{D}_i\boldsymbol{V}_i^{\mathrm{T}}$. This further implies that

$$\boldsymbol{C} = \left(\bigotimes_{i=1}^{n} \boldsymbol{U}_i\right)\left(\bigotimes_{i=1}^{n} \boldsymbol{D}_i\right)\left(\bigotimes_{i=1}^{n} \boldsymbol{V}_i\right)^{\mathrm{T}}.$$

Observe that both $\bigotimes_{i=1}^{n} \boldsymbol{U}_i$ and $\bigotimes_{i=1}^{n} \boldsymbol{V}_i$ correspond to practically implementable single-qubit gates. Since the Kronecker product preserves the triangular structure, we see that $\bigotimes_{i=1}^{n} \boldsymbol{D}_i$ also represents a triangular channel.

---

[1] The term “coherent channels” refers to the channels having unitary matrix representations and themselves can be implemented using (error-free) unitary gates.

### 4.2.2 Analysis on Triangular Channels

According to the discussions in Section 4.2.1, we are particularly interested in the quasi-probability representation of triangular channels, whose Pauli transfer matrices take the same form as the matrix $\boldsymbol{D}$ in (4.14). More precisely, we define triangular channels as CPTP quantum channels whose Pauli transfer matrix can be written as follows

$$\boldsymbol{D} = \begin{bmatrix} 1 & \boldsymbol{0}^{\mathrm{T}} \\ \boldsymbol{b} & \boldsymbol{L} \end{bmatrix}, \tag{4.16}$$

where $\boldsymbol{L}$ is a lower triangular matrix. This includes both amplitude damping channels and Pauli channels as representative examples. For example, a single-qubit amplitude damping channel having decay probability $p$ has the following Pauli transfer matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{1-p} & 0 & 0 \\ 0 & 0 & \sqrt{1-p} & 0 \\ p & 0 & 0 & 1-p \end{bmatrix},$$

where the rows/columns are ordered for ensuring that they correspond to the Pauli-I, X, Y, and Z operators, respectively. Observe that the matrix does have a triangular structure, which is preserved under the permutation of the Pauli-X, Y, and Z operators. As a direct corollary, for a multi-qubit channel inflicting amplitude damping independently on each qubit, the Pauli transfer matrix is triangular, since the triangular structure is preserved under the Kronecker product.

For a fair comparison, we consider channels having the same GGEP $\epsilon$, meaning that $\boldsymbol{D} \in \mathscr{C}_n(\epsilon)$, where we have:

$$\mathscr{C}_n(\epsilon) = \{\boldsymbol{D} \in \mathscr{C}_n |\, 4^{-n}\mathrm{Tr}\{\boldsymbol{D}\} = 1-\epsilon\}, \tag{4.17}$$

and $\mathscr{C}_n$ denotes the set of Pauli transfer matrices of all CPTP triangular channels over $n$ qubits. A noteworthy fact is that $\boldsymbol{D}$ is a lower-triangular matrix, which will be exploited in our forthcoming discussions.

In the following proposition, we will show that regardless of the specific choice of the basis $\boldsymbol{B}$, Pauli channels have the lowest SOF among all triangular CPTP channels. The Pauli channels are defined as channels that transform one Pauli matrix into another. Based on (2.17), this implies that the corresponding Pauli transfer matrices are diagonal matrices.

**Proposition 4.3** (Pauli channels have the lowest SOF)**.** *Given a fixed GGEP $\epsilon$, for any full-rank basis matrix $\boldsymbol{B}$ consisting of vectorized Pauli transfer matrix representation of CPTnI operators, among all CPTP channels over $n$ qubits whose Pauli transfer matrix representation is a triangular matrix, Pauli channels have the lowest SOF.*

*Proof.* Please refer to Appendix A.1. □

*Remark* 4.4. We note that there is some empirical evidence supporting that projecting a quantum channel onto the set of Pauli channels might help in reducing the sampling overhead [38]. Here we formally show that this is indeed true, and it is true for all channels having triangular Pauli transfer matrices, namely for the family of triangular channels.

Next, we consider some notational simplifications for a further investigation in the context of Pauli channels. First of all, since the Pauli transfer matrices of Pauli channels are diagonal (as exemplified by (3.38)), we may rewrite the vector representation of a Pauli transfer matrix (or that of its inverse) in a reduced-dimensional manner. Specifically, we could represent the Pauli transfer matrix of a Pauli channel using merely the vector on its main diagonal, i.e. $\boldsymbol{d} = \text{vec}\{\text{mdiag}\{\boldsymbol{D}\}\}$. Hence we have $\boldsymbol{D}\boldsymbol{x} = \boldsymbol{d} \circ \boldsymbol{x}$.

Additionally, the $16^n \times 16^n$ basis matrix $\boldsymbol{B}$ can be reduced to $\tilde{\boldsymbol{B}} \in \mathbb{R}^{4^n \times 4^n}$ for Pauli channels. In light of this, we have

$$\boldsymbol{\mu}_{\mathcal{D}} = \text{vec}\{\text{mdiag}\{\tilde{\boldsymbol{B}}^{-1}(1/\boldsymbol{d})\}\}. \tag{4.18}$$

For simplicity, we introduce the further notation of

$$\tilde{\boldsymbol{\mu}}_{\mathcal{D}} = \tilde{\boldsymbol{B}}^{-1}(1/\boldsymbol{d}). \tag{4.19}$$

Note that each column in $\tilde{\boldsymbol{B}}$ represents the vectorized Pauli transfer matrix of a specific Pauli operator (as a quantum channel). According to the definitions of the Pauli operators and (2.17), under the computational basis, the vectorized Pauli transfer matrix representation of single-qubit Pauli operators can be expressed as

$$\begin{aligned} \boldsymbol{s}_{\mathcal{I}} &= [1\,1\,1\,1]^{\mathrm{T}}, \\ \boldsymbol{s}_{\mathcal{X}} &= [1\,1\,-1\,-1]^{\mathrm{T}}, \\ \boldsymbol{s}_{\mathcal{Y}} &= [1\,-1\,1\,-1]^{\mathrm{T}}, \\ \boldsymbol{s}_{\mathcal{Z}} &= [1\,-1\,-1\,1]^{\mathrm{T}}, \end{aligned} \tag{4.20}$$

respectively. In this case, it can be seen that the corresponding simplified basis matrix of

$$\tilde{\boldsymbol{B}}_1 = [\boldsymbol{s}_{\mathcal{I}}\ \boldsymbol{s}_{\mathcal{X}}\ \boldsymbol{s}_{\mathcal{Y}}\ \boldsymbol{s}_{\mathcal{Z}}]$$

has the form of the Hadamard transform matrix. In general, Pauli operators over $n$ qubits can be expressed as the tensor product of $n$ single-qubit Pauli operators, hence the corresponding $\tilde{\boldsymbol{B}}$ takes the form of

$$\tilde{\boldsymbol{B}} = (\tilde{\boldsymbol{B}}_1)^{\otimes n},$$

which is simply a Hadamard transform matrix of higher dimensionality, where $\tilde{\boldsymbol{B}}_1$ denotes the matrix $\tilde{\boldsymbol{B}}$ for single-qubit systems.

By exploiting the properties of the Hadamard transform, we are now able to obtain the following result.

**Proposition 4.5** (Depolarizing channels have the lowest SOF)**.** *Among all Pauli channels over n qubits having the GGEP $\epsilon$, depolarizing channels have the lowest SOF.*

*Proof.* Please refer to Appendix A.2. □

According to Proposition 4.5, depolarizing channels lend themselves most readily to be compensated by CI-QEM. This means that the CI-QEM method has a strikingly different nature compared to the family of quantum error correction schemes in terms of the overhead imposed, since depolarizing channels may be viewed as the channels most impervious for QECCs. To elaborate, depolarizing channels exhibit the lowest hashing bound among all Pauli channels, hence they would require the highest qubit overhead[2] for QECCs [48].

### 4.2.3 Bounding the SOF of Pauli Channels

In Section 4.2.2 we have shown that Pauli channels are preferable for CI-QEM in the sense that they have the lowest SOF. In this subsection, we proceed by further investigating Pauli channels and bound their SOF for a given GGEP $\epsilon$. First of all, by explicitly calculating the SOF of depolarizing channels, we can readily obtain a lower bound on the SOF of triangular channels (hence also on Pauli channels), as stated below.

**Corollary 4.6** (SOF lower bound)**.** *For an triangular channel $\mathcal{C}$ having the GGEP $\epsilon$, the SOF incurred by CI-QEM is bounded from below as*

$$\gamma_{\mathcal{C}} \geq 4\epsilon \cdot \frac{1}{(1-\epsilon)^2}. \tag{4.21}$$

*The lower bound is attained, when $\mathcal{C}$ is a depolarizing channel.*

*Proof.* According to Proposition 4.5, it is clear that the channel having the lowest SOF among all $N$-qubit triangular channels is the $N$-qubit depolarizing channel. The SOF of this channel is given by

$$\begin{aligned} \gamma &= \left\| \boldsymbol{H}_N^{-1} \left( 1/ \left( \boldsymbol{H}_N \boldsymbol{d}_N \right) \right) \right\|_1^2 - 1 && \text{(4.22a)} \\ &= \left( \frac{(4^N-1)(1-2\epsilon)-\epsilon}{4^N(1-\epsilon)-1} \right)^2 - 1. && \text{(4.22b)} \end{aligned}$$

---

[2] Given a fixed GEP.

where $\boldsymbol{H}_N$ and $\boldsymbol{H}_N^{-1}$ are the Hadamard transform matrix and the inverse Hadamard transform matrix having dimensionality of $16^N \times 16^N$, respectively, and we have $\boldsymbol{d}_N = [1-\epsilon\ \epsilon(4^N-1)^{-1}\mathbf{1}^{\mathrm{T}}]$. It is straightforward to verify that the right hand side of (4.22b) is a monotonically decreasing function of $N$, thus

$$\gamma \leq \lim_{N\to\infty}\left(\frac{(4^N-1)(1-2\epsilon)-\epsilon}{4^N(1-\epsilon)-1}\right)^2 - 1 \tag{4.23a}$$

$$= \frac{4\epsilon}{(1-\epsilon)^2}. \tag{4.23b}$$

Hence the proof is completed. □

Let us now derive an upper bound on the SOF of Pauli channels. For this purpose, we consider a matrix representation specifically designed for Pauli channels, which will be referred to as the Pauli random walk (PRW) representation hereafter. More precisely, a Pauli channel $\mathcal{C}$ over $n$ qubits can be represented by a matrix $\boldsymbol{C}_{\text{PRW}}$ having the following form:

$$[\boldsymbol{C}_{\text{PRW}}]_{i,j} = \frac{1}{2^n}\mathrm{Tr}\{\mathcal{P}_i\mathcal{C}\mathcal{P}_j\}. \tag{4.24}$$

Conventionally, an $n$-qubit Pauli channel $\mathcal{C}$ can be characterized using a vector $\boldsymbol{\eta}_{\mathcal{C}}$ satisfying

$$\mathcal{C}(\rho) = \sum_{i=1}^{4^n}[\boldsymbol{\eta}_{\mathcal{C}}]_i \boldsymbol{S}_i^{(n)}\rho\left(\boldsymbol{S}_i^{(n)}\right)^{\dagger}. \tag{4.25}$$

We will refer to $\boldsymbol{\eta}_{\mathcal{C}}$ as the probability vector of $\mathcal{C}$ in the rest of this paper. For examples of $\boldsymbol{\eta}_{\mathcal{C}}$ values corresponding to some simple channel models, please refer to Appendix B.1. In light of (4.25), the PRW representation can be expressed as a function of $\boldsymbol{\eta}_{\mathcal{C}}$ as follows

$$[\boldsymbol{C}_{\text{PRW}}(\boldsymbol{\eta}_{\mathcal{C}})]_{i,j} = [\boldsymbol{\eta}_{\mathcal{C}}]_l,$$
$$\mathcal{P}_i\mathcal{P}_j = \mathcal{P}_l.$$

To gain further insights into the PRW representation, we will rely on a weighted Cayley graph [155] $\mathscr{G}$ of Pauli groups, in which the $i$-th vertex represents the $i$-th operator in $\mathscr{P}^n$. For a specific channel $\mathcal{C}$, a pair of nodes $i$ and $j$ in the Cayley graph are connected with an edge having a weight of $[\boldsymbol{\eta}_{\mathcal{C}}]_l$, if we have $\mathcal{P}_i\mathcal{P}_j = \mathcal{P}_l$. As a tangible example, the graph $\mathscr{G}$ corresponding to the single-qubit Pauli group is portrayed in Fig. 4.2. The function $\sigma(\mathcal{O})$ denotes the index of the operator $\mathcal{O}$ in $\mathscr{P}$, where we have $\sigma(\mathcal{X}) = 2$, $\sigma(\mathcal{Y}) = 3$, and $\sigma(\mathcal{Z}) = 4$. For a fixed GGEP $\epsilon$, we can rewrite $\boldsymbol{C}_{\text{PRW}}(\boldsymbol{\eta}_{\mathcal{C}})$ as

$$\boldsymbol{C}_{\text{PRW}}(\boldsymbol{\eta}_{\mathcal{C}}) = (1-\epsilon)\boldsymbol{I} + \boldsymbol{A}(\mathscr{G},\boldsymbol{\eta}_{\mathcal{C}}), \tag{4.26}$$

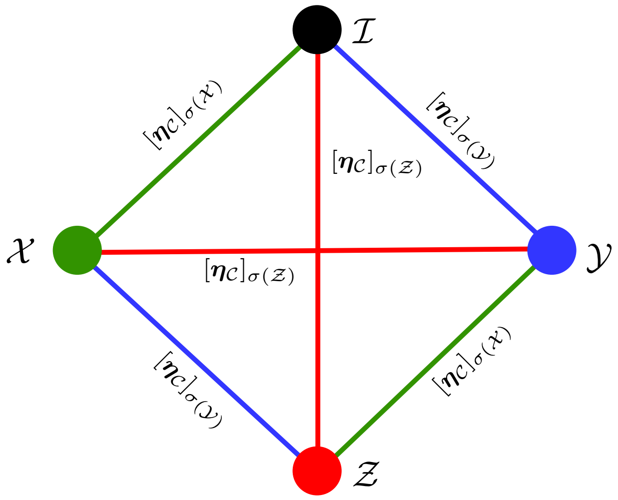


FIGURE 4.2: The weighted Cayley graph $\mathscr{G}$ of the single-qubit Pauli group $\mathscr{P}$ constructed based on channel $\mathcal{C}$. The vertices represent the four operators in $\mathscr{P}$. The quantities next to the edges represent the corresponding edge weights.

where $\boldsymbol{A}(\mathscr{G}, \boldsymbol{\eta}_{\mathcal{C}})$ is the weighted adjacency matrix of the graph $\mathscr{G}$ corresponding to the channel $\mathcal{C}$, which satisfies

$$[\boldsymbol{A}(\mathscr{G}, \boldsymbol{\eta}_{\mathcal{C}})]_{i,j} = \begin{cases} [\boldsymbol{\eta}_{\mathcal{C}}]_l, & i \neq j,\ \mathcal{P}_i\mathcal{P}_j = \mathcal{P}_l; \\ 0, & i = j. \end{cases} \tag{4.27}$$

Whenever there is no confusion, we will simply denote $\boldsymbol{C}_{\text{PRW}}(\boldsymbol{\eta}_{\mathcal{C}})$ as $\boldsymbol{C}$, and $\boldsymbol{A}(\mathscr{G}, \boldsymbol{\eta}_{\mathcal{C}})$ as $\boldsymbol{A}$.

It can be observed from (4.27) that the channel $\mathcal{C}$ may be interpreted as a random walk over the graph $\mathscr{G}$, which maps an input state $|\psi\rangle\langle\psi|$ to $\mathcal{P}_i|\psi\rangle\langle\psi|$ with probability $\eta_i$. The goal of the quasi-probability representation method is to find another operator that reverses the random walk process. Specifically, (3.31) can be simplified as follows

$$\tilde{\boldsymbol{\mu}}_{\mathcal{C}} = \boldsymbol{C}^{-1}\boldsymbol{\alpha}, \tag{4.28}$$

where $\boldsymbol{\alpha} = [1\ \boldsymbol{0}_{4^n-1}^{\mathrm{T}}]^{\mathrm{T}}$, and $\tilde{\boldsymbol{\mu}}_{\mathcal{C}}$ is obtained by extracting the $4^n$ entries corresponding to Pauli operators from $\boldsymbol{\mu}_{\mathcal{C}}$ in (3.31).

With the aid of PRW representation, we are now ready to present the following SOF upper bound for Pauli channels.

**Proposition 4.7** (SOF upper bound)**.** *For an $n$-qubit Pauli channel $\mathcal{C}$, given a GGEP $\epsilon$, the SOF can be upper bounded as*

$$\gamma_{\mathcal{C}} \leq 4\epsilon \cdot \frac{1-\epsilon}{(1-2\epsilon)^2}. \tag{4.29}$$

*The equality is attained when there is only a single type of error, namely there is only one non-zero entry in $\boldsymbol{\eta}_{\mathcal{C}}$.*

*Proof.* Please refer to Appendix A.3. □

Note that Pauli channels having only a single type of error correspond to the highest hashing bound, when mitigated by QECCs. Therefore, by considering both Corollary 4.6 and Proposition 4.7, one may intuitively conjecture that for Pauli channels having the same GGEP, the SOF increases as the hashing bound increases. The hashing bound of a Pauli channel $\mathcal{C}$ can be expressed as [156–158]

$$R_{\text{hashing}} = 1 - H(\boldsymbol{\eta}_{\mathcal{C}}), \tag{4.30}$$

where $R_{\text{hashing}}$ is the highest affordable coding rate capable of satisfying the hashing bound, and $H(\boldsymbol{\eta}_{\mathcal{C}})$ denotes the entropy of $\boldsymbol{\eta}_{\mathcal{C}}$ viewed as a probability distribution. Mathematically, the entropy $H(\boldsymbol{\eta}_{\mathcal{C}})$ is a Schur-concave function [159, Section 2.1] with respect to the probability distribution $\boldsymbol{\eta}_{\mathcal{C}}$. To elaborate, a Schur-concave function $f(\boldsymbol{x})$ is characterized by

$$f(\boldsymbol{x}) \leq f(\boldsymbol{Q}\boldsymbol{x}),$$

for any doubly stochastic matrix $\boldsymbol{Q}$. This implies that doubly stochastic transformations on $\boldsymbol{\eta}_{\mathcal{C}}$ would lead to the increase of entropy [160]. The term $R_{\text{hashing}}$ in (4.30) can be seen to have the exactly opposite property termed as Schur-convex [159, Section 2.1], hence the aforementioned conjecture can be formulated as follows: "the SOF is a Schur-convex function with respect to the probability vector $\boldsymbol{\eta}_{\mathcal{C}}$".

Next we show that the conjecture is correct, when the channels under consideration are memoryless channels, in the sense that they can be described by the tensor product of single-qubit channels.

**Proposition 4.8.** *For any n-qubit memoryless Pauli channel $\mathcal{C} = \bigotimes_{i=1}^{n} \mathcal{C}_i$, given a fixed GGEP $\epsilon$, the SOF is a Schur-convex function of $\boldsymbol{\eta}_{\mathcal{C}}$, meaning that*

$$\|\boldsymbol{\mu}(\boldsymbol{\eta}_{\mathcal{C}})\|_1 \geq \left\| \boldsymbol{\mu}\left( \bigotimes_{i=1}^{n} \boldsymbol{Q}_i \boldsymbol{\eta}_{\mathcal{C}_i} \right) \right\|_1 \tag{4.31}$$

*holds for all doubly-stochastic matrices $\boldsymbol{Q}_i$ preserving the GGEP, where $\boldsymbol{\mu}(\boldsymbol{x})$ denotes the quasi-probability representation vector of the Pauli channel having the probability vector of $\boldsymbol{x}$.*

*Proof.* Please refer to Appendix A.4. □

### 4.2.4 SOF Reduction Using Quantum Channel Precoders: Practical Considerations

Our previous analysis indicates that depolarizing channels are the most preferable channels in terms of having the lowest SOF. This implies that Clifford twirling [126,

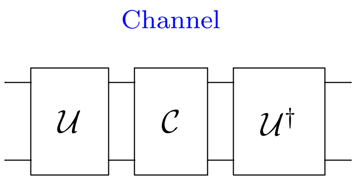


FIGURE 4.3: Clifford twirling.

161, 162], a technique that turns an arbitrary channel into a depolarizing channel while preserving the original average fidelity, might be a quantum channel precoder enabling effective SOF reduction. Specifically, given a quantum channel $\mathcal{C}$ over $n$ qubits, the Clifford twirling $\mathcal{T}_{\mathscr{C}}$ transforms the channel such that the output state satisfies

$$\mathcal{T}_{\mathscr{C}}[\mathcal{C}(\rho)] = \sum \boldsymbol{U}^\dagger \mathcal{C}(\boldsymbol{U}\rho\boldsymbol{U}^\dagger)\boldsymbol{U}, \tag{4.32}$$

where the summation is carried out over the Clifford group on $n$ qubits. Conceptually, the Clifford twirling over two-qubit channels can be implemented as demonstrated in Fig. 4.3, where the gates comprising the circuits $\mathcal{U}$ and $\mathcal{U}^\dagger$ are chosen according to a uniform distribution over the set of Clifford gates.

In practice, however, the gates used for implementing Clifford twirling might be imperfect themselves. In light of this, a real-world Clifford twirling would in general impose an average fidelity reduction, and thus lead to additional SOF. For certain channels, the theoretical SOF reduction of Clifford twirling may be outweighed by this additional overhead. A representative example is constituted by the family of Pauli channels, whose SOF is rather close to that of depolarizing channels, according to Proposition 4.7.

The observation that the Pauli channels have similar SOFs implies that Pauli twirling $\mathcal{T}_{\mathscr{P}}$ might be a more practical quantum channel precoder, which turns an arbitrary channel into a Pauli channel in the following manner [125]

$$\mathcal{T}_{\mathscr{P}}[\mathcal{C}(\rho)] = \frac{1}{4^n}\sum_{i=1}^{4^n} \boldsymbol{S}_i^\dagger \mathcal{C}(\boldsymbol{S}_i\rho\boldsymbol{S}_i^\dagger)\boldsymbol{S}_i. \tag{4.33}$$

The implementation of Pauli twirling is portrayed in Fig. 4.4, where gates $A$, $B$, $C$ and $D$ are chosen according to a uniform distribution on the set of Pauli gates. In state-of-the-art quantum computers, two-qubit gates, as used in the Clifford twirling shown in Fig. 4.3, would result in much more error than single-qubit gates (typically by a factor of 10 or even higher [163]), hence Pauli twirling using single-qubit gates may introduce much lower additional SOF than Clifford twirling.

In practice, we cannot directly implement twirling at both sides of the channel. Instead, we have to twirl simultaneously both the perfect gate and the channel. Therefore, the

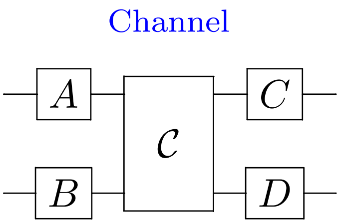


FIGURE 4.4: Pauli twirling.

techniques should be slightly modified in order to effectively apply the twirling to the channel. Specifically, if we wish to apply Pauli twirling to a channel $\mathcal{C}$ associated with an imperfect gate $\tilde{\mathcal{G}} = \mathcal{C} \circ \mathcal{G}$ where $\mathcal{G}$ denotes the perfect gate, we may apply the following modified twirling to $\tilde{\mathcal{G}}$

$$\begin{aligned}\widetilde{\mathcal{T}}_{\mathscr{P}}[\tilde{\mathcal{G}}(\rho)] &= \frac{1}{4^n}\sum_{i=1}^{4^n} \boldsymbol{S}_i^\dagger \tilde{\mathcal{G}}[(\boldsymbol{G}^\dagger \boldsymbol{S}_i \boldsymbol{G})\rho(\boldsymbol{G}^\dagger \boldsymbol{S}_i \boldsymbol{G})^\dagger]\boldsymbol{S}_i \\ &= \frac{1}{4^n}\sum_{i=1}^{4^n} \boldsymbol{S}_i^\dagger \mathcal{C}[\boldsymbol{S}_i \boldsymbol{G}\rho \boldsymbol{G}^\dagger \boldsymbol{S}_i^\dagger]\boldsymbol{S}_i \\ &= [\mathcal{T}_{\mathscr{P}}(\mathcal{C}) \circ \mathcal{G}](\rho).\end{aligned}$$

A similar procedure can also be applied to Clifford twirling. We note that the operation $(\boldsymbol{G}^\dagger \boldsymbol{S}_i \boldsymbol{G})\rho(\boldsymbol{G}^\dagger \boldsymbol{S}_i \boldsymbol{G})^\dagger$ can be simplified to $\boldsymbol{S}_m \rho \boldsymbol{S}_m^\dagger$ for some $m$, if the perfect gate $\mathcal{G}$ is a Clifford gate, since the Clifford group $\mathscr{C}_n$ is the normalizer of the Pauli group $\mathscr{P}_n$ satisfying $\mathscr{C}_n = \{\boldsymbol{U} \in \mathbb{U}(2^n) | \boldsymbol{U}\mathscr{P}_n\boldsymbol{U}^\dagger = \mathscr{P}_n\}$.

## 4.3 Sampling Overhead Factor Analysis for Coded Quantum Gates

In this section, we investigate the SOF of gates protected by quantum channel codes, including QECCs and QEDCs. These codes are designed to convert the original channel corresponding to the unprotected gate into an reduced-error-rate channel over more qubits, with the objective of having lower GGEP. Under the framework of CI-QEM, they can also be viewed as channel precoders. Naturally, it is of great interest to us, whether an amalgam of quantum codes and CI-QEM can benefit each other.

Specifically, we consider the scenario where every set of $k$ logical qubits is protected using $n$ physical qubits. Using the terminology of quantum coding, this means that we consider $[[n, k, d]]$ codes, where $d$ is the minimum distance of the code [26]. Furthermore, if not otherwise stated, we assume that Clifford gates are considered using the transversal gate scheme of [44], while non-Clifford gates are implemented via the magic state distillation process of [164]. These are conventional assumptions in the quantum

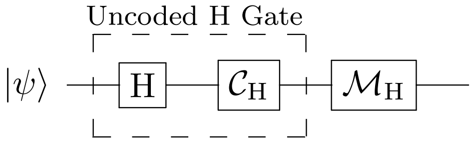


FIGURE 4.5: Conceptual schematic of a CI-QEM-protected uncoded Hadamard gate.

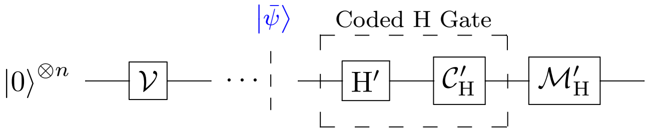


FIGURE 4.6: Conceptual schematic of a CI-QEM-protected coded Hadamard gate.

fault-tolerant computing literature [48, Section 10.6]. Furthermore, for the conciseness of discussion, we assume that the quantum channels encountered in this section are all Pauli channels, or had been turned into Pauli channels by means of Pauli twirling.

### 4.3.1 Amalgamating Quantum Codes with CI-QEM: A Toy Example

To elaborate further on how CI-QEM may be amalgamated with quantum codes, we consider the simple example of protecting a single Hadamard gate. As shown in Fig. 4.5, an uncoded imperfect Hadamard gate can be decomposed into a perfect Hadamard gate H and a quantum channel $\mathcal{C}_{\mathrm{H}}$. Given that the channel $\mathcal{C}_{\mathrm{H}}$ is known, we can apply the CI-QEM circuit $\mathcal{M}_{\mathrm{H}}$ to invert it. By contrast, in the coded scheme, the logical qubit is protected using an encoder $\mathcal{V}$ exploiting $n$ physical qubits at the input of the circuit, as portrayed in Fig. 4.6. In the code space, the original input state $|\psi\rangle$ is expressed as the coded state $|\bar{\psi}\rangle$, while the coded Hadamard gate may be decomposed into an equivalent perfect Hadamard gate H′ and another quantum channel $\mathcal{C}'_{\mathrm{H}}$. Consequently, the CI-QEM circuit $\mathcal{M}'_{\mathrm{H}}$ has to be designed for the transformed channel $\mathcal{C}'_{\mathrm{H}}$. More specifically, the Hadamard gate protected using the transversal gate configuration is depicted in more detail in Fig. 4.7. The equivalent Hadamard gate is implemented simply by $n$ transversal Hadamard gates. As a result, each physical qubit experiences the same channel $\mathcal{C}_{\mathrm{H}}$. Right after the transversal gates, with the help of $m$ ancillae, the integrity of the output state is examined by the stabilizer check $\mathcal{S}$. The subsequent recovery circuit $\mathcal{R}$ is capable of correcting a fixed number of Pauli errors, depending on the minimum distance of the code. For example, if Steane's codes is applied, $\mathcal{R}$ can correct any single Pauli error that appeared within the circuit. The transversal gates along with the stabilizer check and the recovery circuit constitute the transformed channel $\mathcal{C}'_{\mathrm{H}}$.

Ideally, since $\mathcal{S}$ and $\mathcal{R}$ are able to correct errors, the transformed channel $\mathcal{C}'_{\mathrm{H}}$ might have a lower GGEP than the original channel $\mathcal{C}_{\mathrm{H}}$. However, this might not be true in

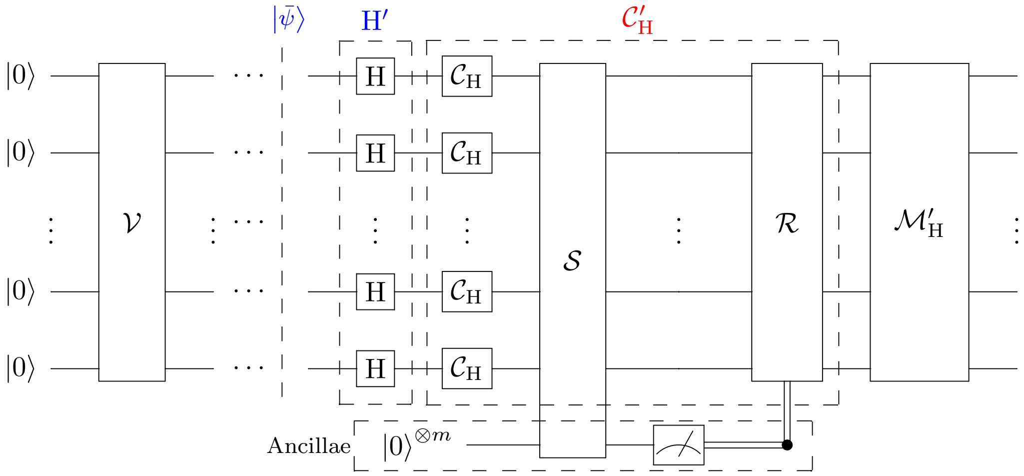


FIGURE 4.7: Detailed schematic of a CI-QEM-protected coded Hadamard gate relying on the transversal gate configuration.

practice, because $\mathcal{S}$ and $\mathcal{R}$ themselves are also prone to errors. Intuitively, assuming that the GGEP of each gate in the circuit is at most $\epsilon$, as $\epsilon$ tends to zero, the GGEP of $\mathcal{C}'_{\mathrm{H}}$ is at most on the order of $O(\epsilon^2)$, since all single errors are corrected. Therefore, quantum codes are capable of reducing the channel GGEP, provided that $\epsilon$ is sufficiently small. Specifically, the value of physical gate GGEP $\epsilon_{\mathrm{th}}$, below which quantum codes become beneficial, is referred to as the fault-tolerance threshold [165].

In general, given a quantum code, the logical gate GGEP would be higher than the physical gate GGEP, when the physical gate GGEP is relatively high. As the physical gate GGEP decreases, it gradually becomes higher than the logical gate GGEP, as sketched in Fig. 4.8 and detailed in [51, Section VI-B]. The physical gate GGEP corresponding to the cross-over point of the two curves is the fault-tolerance threshold of the quantum code. We will refer to the region where the quantum code is beneficial, in the sense that the logical gate GGEP is lower than the physical gate GGEP, as the error-resilient region, while the opposite region will be referred to as the error-proliferating region.

In the following subsections, we will first analyse the SOF of coded gates when the code is operating in its error-proliferation region, followed by the opposite scenario.

### 4.3.2 Quantum Codes Operating in Their Error-Proliferating Regions

Using our previous results on uncoded gates in Section 4.2.3, it may be readily shown that quantum codes operating in their error-proliferating regions may not lead to substantial SOF reduction. Formally, we have the following result.

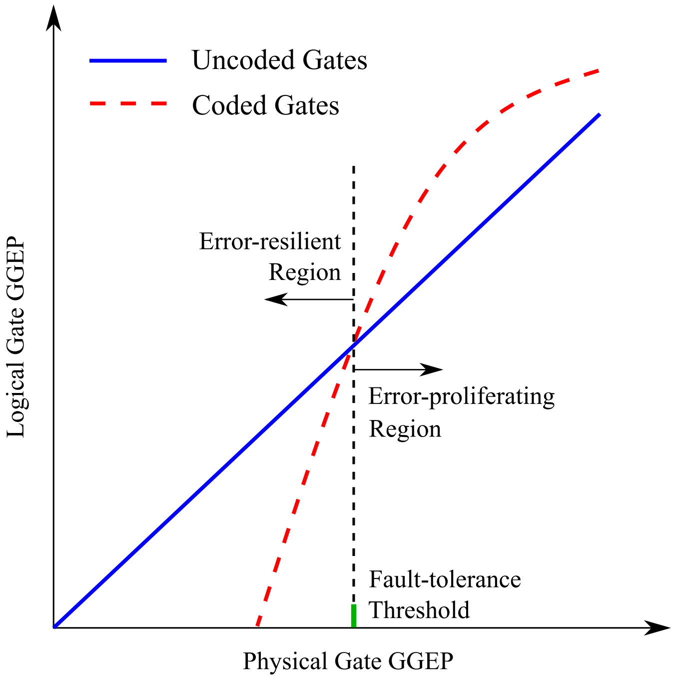


FIGURE 4.8: Illustration of the fault-tolerance threshold, the error-proliferating region and the error-resilient region of quantum codes.

**Corollary 4.9.** *For an uncoded gate having a GGEP of* $\epsilon$ *and SOF* $\gamma$*, the SOF is lower-bounded by* $\gamma \cdot \frac{(1-2\epsilon)^2}{(1-\epsilon)^3}$*, when the gate is protected by some quantum code operating in its error-proliferating region. Furthermore, provided that the channel corresponding to the uncoded gate is a depolarizing channel, the lower bound can be further refined to* $\gamma \cdot \frac{(3-4\epsilon)^2}{3(1-\epsilon)^2(3-\epsilon)}$.

*Proof.* This is a direct corollary following from Corollary 4.6 and Proposition 4.7. To elaborate a little further, considering the extreme case where the threshold is met exactly so that the output GGEP of the quantum code is equal to $\epsilon$, the generic lower bound is obtained in the form of:

$$\gamma \cdot \frac{1}{(1-\epsilon)^2} \cdot \frac{(1-2\epsilon)^2}{1-\epsilon} = \gamma \cdot \frac{(1-2\epsilon)^2}{(1-\epsilon)^3} \tag{4.34}$$

using (4.21) and (4.29). The lower-bound valid for depolarizing channels is obtained as

$$\gamma \cdot \frac{4\epsilon \frac{1}{(1-\epsilon)^2}}{\left(\frac{(4-1)(1-2\epsilon)-\epsilon}{4(1-\epsilon)-1}\right)^2 - 1} = \gamma \cdot \frac{(3-4\epsilon)^2}{3(1-\epsilon)^2(3-\epsilon)} \tag{4.35}$$

using (4.21) and (4.22b), and by further exploiting the fact that single-qubit depolarizing channels have the highest SOF among all depolarizing channels sharing the same GGEP. ☐

To demonstrate the implications of Corollary 4.9 more explicitly, we plot the lower bounds in Fig. 4.9. Since the fault-tolerance thresholds of most QECCs are as low as

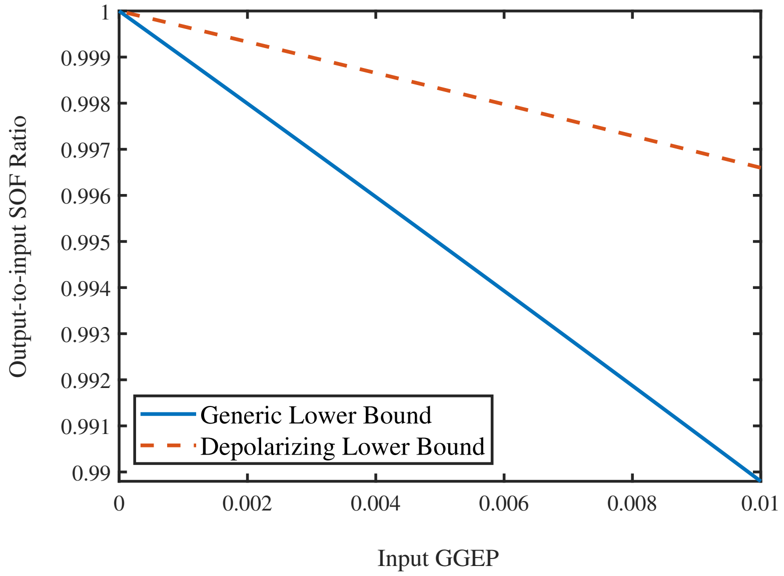


FIGURE 4.9: Lower bounds of the output-to-input SOF ratio for quantum codes operating in their error-proliferating regions.

$10^{-2} \sim 10^{-3}$, as it can be observed from Fig. 4.9, even when the GGEP meets the threshold exactly, the quantum codes can only offer an overhead reduction of at most 1%. Therefore, amalgamating CI-QEM with codes operating in their error-proliferating regions may not be mutually beneficial.

### 4.3.3 QECCs Operating in Their Error-Resilient Regions

In light of our previous discussions, it becomes plausible that QECCs operating in their error-resilient regions may contribute to SOF reduction by reducing the gate error probability. As is known in the literature [48, Section 10.6.1], their error-correcting capability can be further improved via concatenation. However, the price of concatenating codes is a drastic increase in the qubit overhead. It is thus interesting to investigate whether the amalgamation of QECC and CI-QEM would outperform pure concatenated QECCs, and if so, in what scenarios.

To elaborate, we consider the simple example of transversal Hadamard gates protected by a rate $1/3$ repetition code, as portrayed in Fig. 4.10 and detailed in [51]. By concatenating the repetition code twice, the number of physical qubits protecting a single logical qubit become three times that of the non-concatenated code. By contrast, if we amalgamate the rate $1/3$ code with CI-QEM, the additional qubits can be used to parallelize the computation, leading to a computational acceleration by a factor of three. In this sense, the QECC-CI-QEM scheme outperform the concatenated scheme, when the SOF of CI-QEM obeys $\gamma_{\mathcal{C}} \leq 2$.

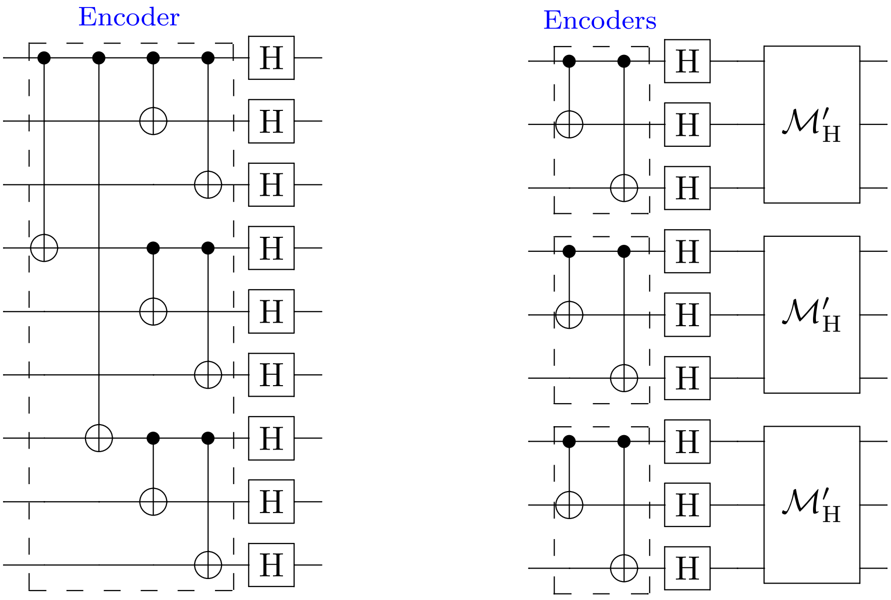


FIGURE 4.10: Comparison between the concatenated QECC scheme (the left figure) and the QECC-CI-QEM scheme (the right figure) protecting transversal Hadamard gates using the rate 1/3 repetition code. In the example, the QECC-CI-QEM scheme runs three times as fast as the concatenation scheme.

As the number of gates in a circuit increases, the CI-QEM sampling overhead grows exponentially (as indicated by (3.37)), while the qubit overhead of the concatenated scheme remains constant. Therefore, there would be a critical point where the overall CI-QEM sampling overhead escalation starts to outweigh the parallelization speedup benefits. This may be interpreted as a limitation imposed on the circuit size, beyond which full fault-tolerance becomes necessary. Next we provide an estimate of the critical point, given that the gate error probability is sufficiently low.

**Proposition 4.10** (Lower bound on the critical point)**.** *Consider a quantum circuit in which each gate has a GGEP at most $\epsilon$. If the gates are protected using $l$-stage concatenated (i.e., concatenated $l$ times) $[[n,k,d]]$ QECC operating in its error-resilient region via the transversal gate configuration, amalgamating the code with CI-QEM is more preferable than applying the $(l+1)$-stage concatenated code, when the number of gates $N_l$ satisfies*

$$N_l \leq \frac{\ln n}{\ln(1+4f^{(l)}(\epsilon)) - \ln[1+4f^{(l+1)}(\epsilon)]} \tag{4.36a}$$

$$\approx \frac{1}{f^{(l)}(\epsilon) - f^{(l+1)}(\epsilon)} \cdot \frac{\ln n}{4} \tag{4.36b}$$

*when $\epsilon \ll 1$, and $f(\epsilon)$ is the output GGEP of the single-stage $[[n,k,d]]$ code given the input GGEP $\epsilon$, and $f^{(l)}(\epsilon)$ denotes the $l$-times self-composition of function $f(\epsilon)$, as exemplified by $f^{(2)}(\epsilon) = f[f(\epsilon)]$.*

*Proof.* For a circuit in which every gate is protected using $l$-stage concatenated $[[n,k,d]]$

QECCs, the total computational overhead (including both the CI-QEM overhead and qubit overhead) $\tilde{\gamma}_l$ can be bounded as

$$\tilde{\gamma}_l \leq n^l (1+\gamma_l)^N, \tag{4.37}$$

where $\gamma_l$ denotes the highest SOF of a single logical gate in the circuit, and $N$ is the total number of gates. Hence the critical point $N_l$ between $l$-stage concatenation and $(l+1)$-stage concatenation satisfies

$$(1+\gamma_l)^{N_l} = n(1+\gamma_{l+1})^{N_l}. \tag{4.38}$$

According to Corollary 4.6 and Propostion 4.7, when $\epsilon \ll 1$, the upper and lower bounds for the SOF of a single gate tend to be equal. Thus the SOF of each uncoded gate can be upper bounded by $\gamma \leq 4\epsilon$. For $l$-stage concatenated codes, we have $\gamma_l \leq 4f^{\circ l}(\epsilon)$. Therefore we obtain

$$N_l \leq \frac{\ln n}{\ln(1+4f^{(l)}(\epsilon)) - \ln[1+4f^{(l+1)}(\epsilon)]}. \tag{4.39}$$

Additionally, using the Maclaurin approximation of $\ln(1+x) \approx x$ when $x > 0$ is sufficiently small, we obtain (4.36b). Hence the proof is completed. □

*Remark* 4.11. As a special case, the pure CI-QEM (i.e., $l = 0$) is more preferable than amalgamating a single-stage $[[n,k,d]]$ code with it, when the number of gates satisfies

$$N_0 \leq \frac{\ln n}{\ln(1+4\epsilon) - \ln[1+4f(\epsilon)]} \tag{4.40a}$$

$$\approx \frac{1}{\epsilon - f(\epsilon)} \cdot \frac{\ln n}{4}. \tag{4.40b}$$

Note that when $(l+1)$-stage QECC concatenation cannot be implemented due to the associated physical limitations (e.g. total number of physical qubits), the amalgam of $l$-stage QECC and CI-QEM may be applied even beyond the critical point.

### 4.3.4 QEDCs Operating in Their Error-Resilient Regions

Due to their smaller minimum distance than that of QECCs, QEDCs are not capable of correcting any error. Nonetheless, they can be used as important building blocks in the scheme of *post-selection fault-tolerance* [166, 167]. To expound a little further, post-selection fault-tolerance differs from its conventional counterpart in that it is implemented by detecting potential errors, and only accepting the results if no error is detected. Typically, QEDCs have a shorter codeword length compared to QECCs, hence they often also possess a higher threshold. For instance, the $[[4,2,2]]$ QEDC can detect an error at the price of protecting a logical gate using four physical gates, while

Steane's $[[7,1,3]]$ QECC requires seven gates. This makes post-selection fault-tolerance a preferable scheme, when the gates are relatively noisy [166–168].

In the context of CI-QEM, the high threshold of QEDCs appears to make their amalgamation with CI-QEM more beneficial. However, a subtle issue is that QEDCs suffer from their own sampling overhead. To elaborate, if for every gate the probability of successful error detection is $p$, similar to that of CI-QEM, the SOF of the QEDC may be defined as

$$\begin{aligned}\gamma_{\mathrm{QEDC}} &= \frac{1}{1-p} - 1 \\ &= \frac{p}{1-p},\end{aligned} \tag{4.41}$$

which will be referred to as the QEDC-SOF in the following disucssions. Additionally, performing CI-QEM on the post-selected channel (where single errors are eliminated) also incurs an SOF, which in this context will be referred to as the CI-QEM-SOF. Thus, the total SOF can be calculated as follows:

$$\begin{aligned}\gamma_{\mathrm{total}} &= (1+\gamma_{\mathrm{QEDC}})(1+\gamma_{\mathrm{QEM}}) - 1 \\ &= \gamma_{\mathrm{QEDC}} + \gamma_{\mathrm{QEM}} + \gamma_{\mathrm{QEDC}} \cdot \gamma_{\mathrm{QEM}},\end{aligned} \tag{4.42}$$

where $\gamma_{\mathrm{QEDC}}$ and $\gamma_{\mathrm{QEM}}$ represent the QEDC-SOF and the CI-QEM-SOF, respectively. In this regard, the amalgamation of QEDC and CI-QEM is only beneficial when the total SOF is lower than that of CI-QEM applied directly to uncoded gates.

To compute the QEDC-SOF of each logical gate for a specific QEDC, it suffices to compute the probability that a single Pauli error occurs in the entire physical circuit corresponding to the logical gate. Let us assume that every single-qubit gate incurs a Pauli channel having a GGEP of $\epsilon_1$, and every two-qubit gate incurs a Pauli channel having GGEP $\epsilon_2$. For any logical gate implemented using the transversal gate configuration, it is clear that the occurrence probability of a single Pauli error in single-qubit and two-qubit gates, namely $p_1$ and $p_2$, can be expressed as

$$p_1 = n\epsilon_1 + O(\epsilon_1^2), \quad p_2 = n\epsilon_2 + O(\epsilon_2^2), \tag{4.43}$$

for any $[[n,k,d]]$ QEDC, since every logical gate is implemented using $n$ physical gates. The terms having the orders of $O(\epsilon_1^2)$ and $O(\epsilon_2^2)$ are negligible when the GEPs are sufficiently small. According to (4.41), we also have the following result for the corresponding QEDC-SOF

$$\gamma_1 = n\epsilon_1 + O(\epsilon_1^2), \quad \gamma_2 = n\epsilon_2 + O(\epsilon_2^2). \tag{4.44}$$

Among QEDCs capable of detecting a single arbitrary Pauli error, the one having the lowest $n$ is the $[[4,2,2]]$ code. In this case, we have $\gamma_1 \approx 4\epsilon_1$ and $\gamma_2 \approx 4\epsilon_2$. The actual QEDC-SOF would be even higher due to the inevitable imperfections in the stabilizer measurements. On the other hand, for small $\epsilon_1$ and $\epsilon_2$, we can see from Corollary 4.6

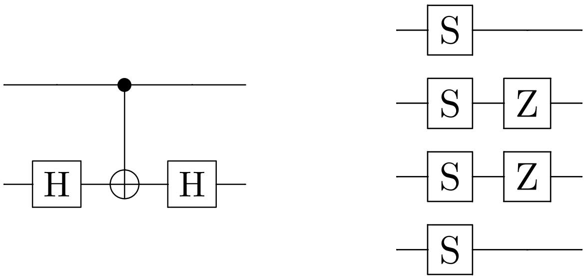


FIGURE 4.11: Illustration of an uncoded controlled-Z gate (the left figure) and a controlled-Z gate protected by the $[[4, 2, 2]]$ QEDC code (the right figure), which is not implemented in a transversal manner.

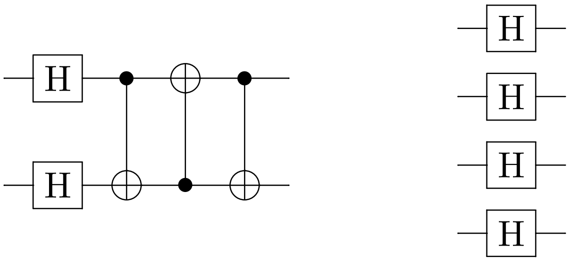


FIGURE 4.12: Illustration of an uncoded $\mathcal{SWAP} \circ \mathcal{H}^{\otimes 2}$ (the left figure) and its coded counterpart protected by the $[[4, 2, 2]]$ QEDC code (the right figure), which is not implemented in a transversal manner.

and Proposition 4.7 that when no QEDC is applied, the CI-QEM-SOF is also approximately four times the GGEP. Therefore, we have the following remark.

*Remark* 4.12. Amalgamating QEDCs and CI-QEM might not be beneficial in the sense of SOF reduction, given that the logical gates are implemented using the transversal gate configuration.

This result may not be applicable to some logical gates, namely to those that are not implemented transversally. For example, some two-qubit gates processing the two logical qubits within a single block of the $[[4, 2, 2]]$ QEDC may be implemented using simple physical gates. As illustrated in Fig. 4.11, a logical controlled-Z gate can be implemented using six single-qubit gates $\mathcal{S} \otimes (\mathcal{Z} \circ \mathcal{S}) \otimes (\mathcal{Z} \circ \mathcal{S}) \otimes \mathcal{S}$. Since two-qubit gates typically have much higher GGEP compared to single-qubit gates, the QEDC-SOF may be even lower than the GGEP of a single controlled-Z gate. Similarly, the logical gate $\mathcal{SWAP} \circ \mathcal{H}^{\otimes 2}$ can be implemented by simply using four physical Hadamard gates, hence also has a low QEDC-SOF. Here, the operator $\mathcal{SWAP}$ refers to the SWAP gate exchanging a pair of qubits [48, Section 1.3.4].

Unfortunately, non-transversal logical gates can only be designed in a case-by-case manner. Moreover, not all of them admit the nice and simple implementation as those shown in Fig. 4.11 and 4.12. For example, a CNOT gate between the two qubits in a

TABLE 4.1
Sixteen basis operators used in CI-QEM for single-qubit channels.

| | Operator | Output state |
|---|---|---|
| 1 | $\mathcal{I}$ | $\mathcal{I}(\rho) = \rho$ |
| 2 | $\mathcal{X}$ | $\mathcal{X}(\rho) = S_{\mathcal{X}}\rho S_{\mathcal{X}}^{\dagger}$ |
| 3 | $\mathcal{Y}$ | $\mathcal{Y}(\rho) = S_{\mathcal{Y}}\rho S_{\mathcal{Y}}^{\dagger}$ |
| 4 | $\mathcal{Z}$ | $\mathcal{Z}(\rho) = S_{\mathcal{Z}}\rho S_{\mathcal{Z}}^{\dagger}$ |
| 5 | $\mathcal{R}_{\mathrm{x}}$ | $\mathcal{R}_{\mathrm{x}}(\rho) = \frac{1}{2}(\boldsymbol{I} + \imath S_{\mathcal{X}})\rho(\boldsymbol{I} + \imath S_{\mathcal{X}})^{\dagger}$ |
| 6 | $\mathcal{R}_{\mathrm{y}}$ | $\mathcal{R}_{\mathrm{y}}(\rho) = \frac{1}{2}(\boldsymbol{I} + \imath S_{\mathcal{Y}})\rho(\boldsymbol{I} + \imath S_{\mathcal{Y}})^{\dagger}$ |
| 7 | $\mathcal{R}_{\mathrm{z}}$ | $\mathcal{R}_{\mathrm{z}}(\rho) = \frac{1}{2}(\boldsymbol{I} + \imath S_{\mathcal{Z}})\rho(\boldsymbol{I} + \imath S_{\mathcal{Z}})^{\dagger}$ |
| 8 | $\mathcal{R}_{\mathrm{yz}}$ | $\mathcal{R}_{\mathrm{yz}}(\rho) = \frac{1}{2}(S_{\mathcal{Y}} + S_{\mathcal{Z}})\rho(S_{\mathcal{Y}} + S_{\mathcal{Z}})^{\dagger}$ |
| 9 | $\mathcal{R}_{\mathrm{xz}}$ | $\mathcal{R}_{\mathrm{xz}}(\rho) = \frac{1}{2}(S_{\mathcal{X}} + S_{\mathcal{Z}})\rho(S_{\mathcal{X}} + S_{\mathcal{Z}})^{\dagger}$ |
| 10 | $\mathcal{R}_{\mathrm{xy}}$ | $\mathcal{R}_{\mathrm{xy}}(\rho) = \frac{1}{2}(S_{\mathcal{X}} + S_{\mathcal{Y}})\rho(S_{\mathcal{X}} + S_{\mathcal{Y}})^{\dagger}$ |
| 11 | $\pi_{\mathrm{x}}$ | $\pi_{\mathrm{x}}(\rho) = \frac{1}{4}(\boldsymbol{I} + S_{\mathcal{X}})\rho(\boldsymbol{I} + S_{\mathcal{X}})^{\dagger}$ |
| 12 | $\pi_{\mathrm{y}}$ | $\pi_{\mathrm{y}}(\rho) = \frac{1}{4}(\boldsymbol{I} + S_{\mathcal{Y}})\rho(\boldsymbol{I} + S_{\mathcal{Y}})^{\dagger}$ |
| 13 | $\pi_{\mathrm{z}}$ | $\pi_{\mathrm{z}}(\rho) = \frac{1}{4}(\boldsymbol{I} + S_{\mathcal{Z}})\rho(\boldsymbol{I} + S_{\mathcal{Z}})^{\dagger}$ |
| 14 | $\pi_{\mathrm{yz}}$ | $\pi_{\mathrm{yz}}(\rho) = \frac{1}{4}(S_{\mathcal{Y}} + \imath S_{\mathcal{Z}})\rho(S_{\mathcal{Y}} + \imath S_{\mathcal{Z}})^{\dagger}$ |
| 15 | $\pi_{\mathrm{xz}}$ | $\pi_{\mathrm{xz}}(\rho) = \frac{1}{4}(S_{\mathcal{X}} + \imath S_{\mathcal{Z}})\rho(S_{\mathcal{X}} + \imath S_{\mathcal{Z}})^{\dagger}$ |
| 16 | $\pi_{\mathrm{xy}}$ | $\pi_{\mathrm{xy}}(\rho) = \frac{1}{4}(S_{\mathcal{X}} + \imath S_{\mathcal{Y}})\rho(S_{\mathcal{X}} + \imath S_{\mathcal{Y}})^{\dagger}$ |

$[[4,2,2]]$ code block has to be implemented via a SWAP gate, which has a high QEDC-SOF. By contrast, the transversal gate configuration is a general design paradigm that can be applied to all logical Clifford gates [44]. In this regard, we may draw the conclusion that QEDC-CI-QEM is only beneficial for certain specific non-transversal gate designs.

## 4.4 Numerical Results

In this section, we augment our discussions throughout previous sections by numerical results. Throughout this section, for single-qubit channels, the basis matrix $\boldsymbol{B}$ used for CI-QEM is constituted by the conventional set of quantum operations listed in Table 4.1 [38]. The geometric interpretation of these operations are further portrayed in Fig. 4.13. The basis operators of CI-QEM for two-qubit channels are constituted by the tensor product of these operators. The operators $\mathcal{R}_{\mathrm{x}}$, $\mathcal{R}_{\mathrm{y}}$, and $\mathcal{R}_{\mathrm{z}}$ represent $\pi/2$ rotations around the x-, y-, and z-axes of the Bloch sphere, respectively, while $\mathcal{R}_{\mathrm{yz}}$, $\mathcal{R}_{\mathrm{xz}}$, and $\mathcal{R}_{\mathrm{xy}}$ represent $\pi$ rotations around the axes determined by the equations $\left\{\begin{array}{l} y = z \\ x = 0 \end{array}\right.$, $\left\{\begin{array}{l} x = z \\ y = 0 \end{array}\right.$, and $\left\{\begin{array}{l} x = y \\ z = 0 \end{array}\right.$, respectively. Similarly, the operators $\pi_{\mathrm{x}}$, $\pi_{\mathrm{y}}$, $\pi_{\mathrm{z}}$, $\pi_{\mathrm{yz}}$, $\pi_{\mathrm{xz}}$, and $\pi_{\mathrm{xy}}$ represent the measurement operations on the corresponding axes.

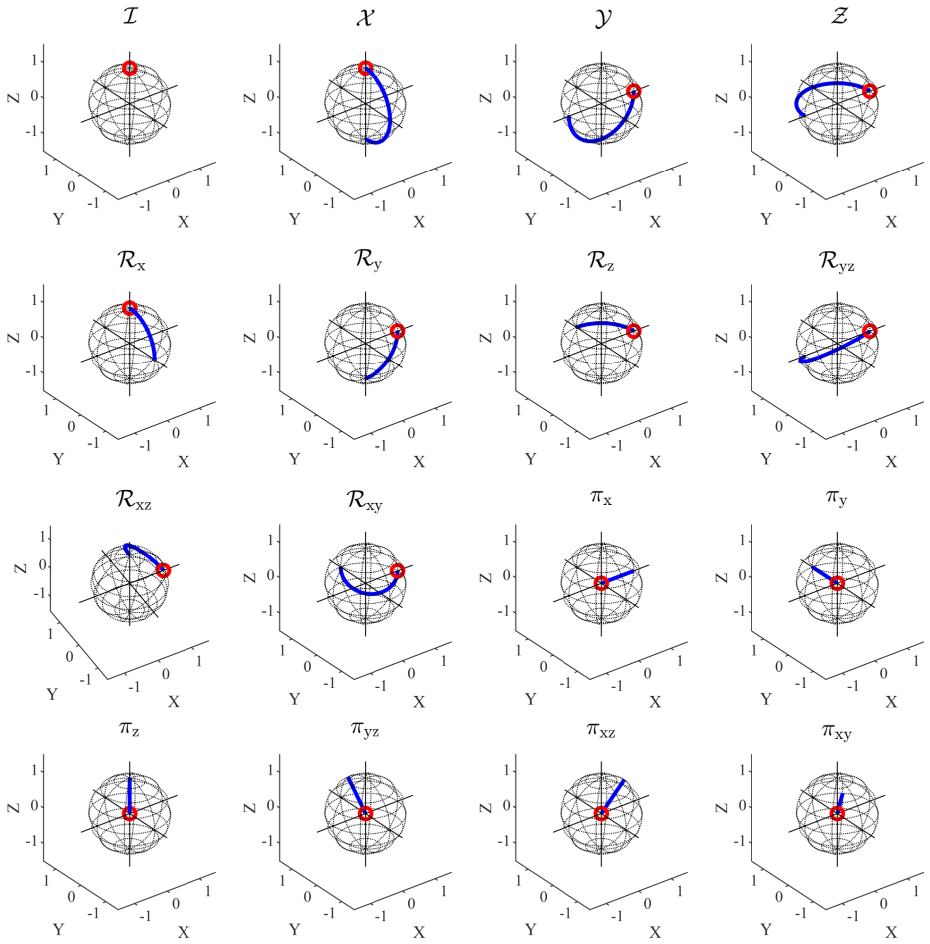


FIGURE 4.13: The trajectories of quantum states under the corresponding basis operations on the Bloch sphere. The circles represent the initial states, while the bold solid lines represent the trajectories.

### 4.4.1 Uncoded Gates

We first characterize Proposition 4.3 and Proposition 4.5 via numerical examples. In Fig. 4.14 and 4.15, the SOF vs. the GGEP is plotted for both single-qubit and two-qubit gates inflicted by coherent errors, amplitude damping and depolarizing channels, as detailed below. Here, the two-qubit channels are restricted to product channels, namely those constructed by the tensor product of two single-qubit channels. Specifically, a single-qubit amplitude damping channel $\mathcal{C}_{damp}$ is characterized by [48, Section 8.3.5]

$$\mathcal{C}_{\mathrm{damp}}(\rho) = \boldsymbol{E}_0\rho\boldsymbol{E}_0^{\dagger} + \boldsymbol{E}_1\rho\boldsymbol{E}_1^{\dagger}, \tag{4.45}$$

where the operation components are given by [48, Section 8.3.5]

$$\boldsymbol{E}_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\delta} \end{bmatrix}, \ \boldsymbol{E}_1 = \begin{bmatrix} 0 & \sqrt{\delta} \\ 0 & 0 \end{bmatrix},$$

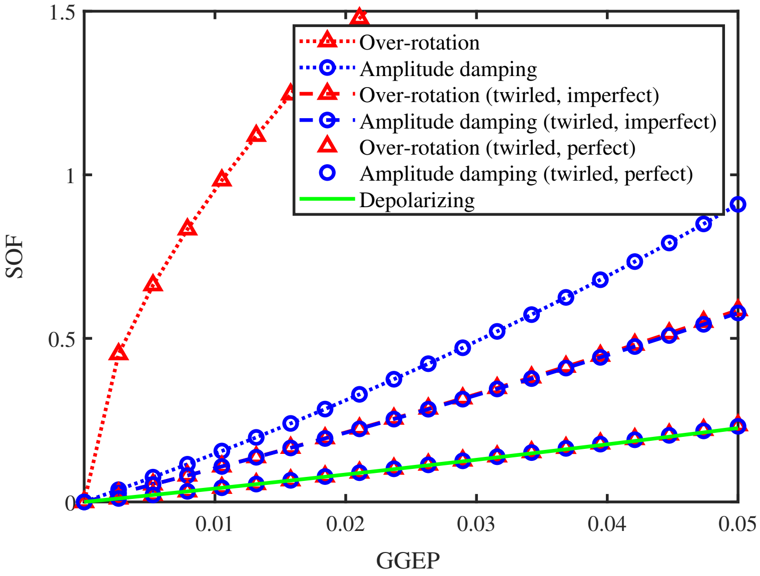


FIGURE 4.14: SOF as a function of GGEP for uncoded single-qubit gates inflicted by coherent error, amplitude damping and depolarizing channels having the same GGEP. The Pauli-twirled versions of coherent error and amplitude damping channels are also plotted for comparison.

and the parameter $\delta$ is the amplitude damping probability of the channel, namely the probability that the channel turns an excited state $|1\rangle$ into the ground state $|0\rangle$. Notably, the amplitude damping channel is a non-Pauli triangular channel. The single-qubit coherent channel we consider here is the over-rotation channel, which takes the form of [38]

$$\mathcal{C}_{\text{over}}(\rho) = \boldsymbol{U}_{\text{x}}\rho\boldsymbol{U}_{\text{x}}^{\dagger}, \tag{4.46}$$

where

$$\boldsymbol{U}_{\text{x}} = \begin{bmatrix} \cos(\frac{4\phi}{\pi}) & \imath\sin(\frac{4\phi}{\pi}) \\ \imath\sin(\frac{4\phi}{\pi}) & \cos(\frac{4\phi}{\pi}) \end{bmatrix}.$$

The parameter $\phi$ controls the over-rotation angle of the channel, and $\imath = \sqrt{-1}$ denotes the imaginary unit.

Observe from Fig. 4.14 and 4.15 that the amplitude damping channels affecting both a single and two qubits have higher SOFs than depolarizing channels. This corroborates Proposition 4.3 and Proposition 4.5, which imply that depolarizing channels have the lowest SOF among all triangular channels. The over-rotation channels are not triangular channels, yet they exhibit the highest SOF. In general, their SOFs would depend on the specific set of basis operators comprised by the matrix $\boldsymbol{B}$. In fact, coherent channels are represented by unitary transformations. In light of this, in the ideal case that "unitary rotation" gates can be implemented without decoherence, they can be compensated in an overhead-free manner by simply applying its complex conjugate.

In Fig. 4.14 and 4.15, we can also see the effect of quantum channel precoders, especially

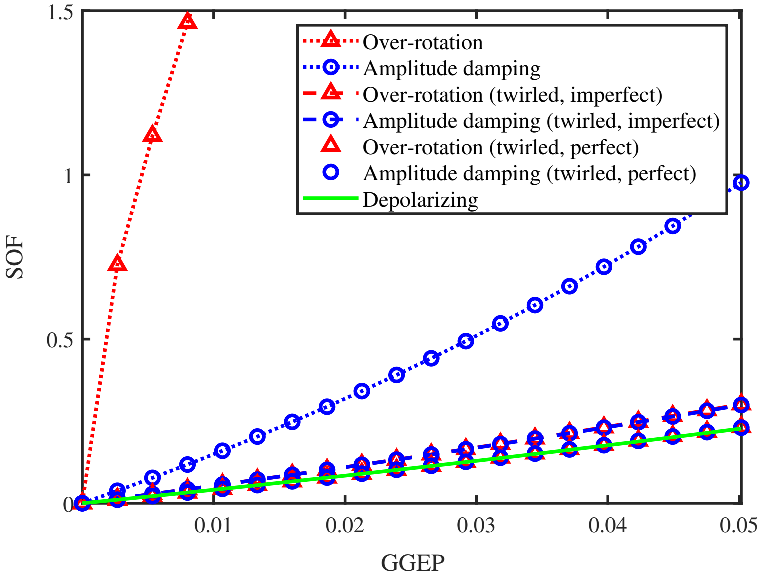


FIGURE 4.15: SOF as a function of GGEP for uncoded two-qubit gates inflicted by coherent error, amplitude damping and depolarizing channels having the same GGEP. The Pauli-twirled versions of coherent error and amplitude damping channels are also plotted for comparison.

that of Pauli twirling. To elaborate, observe in both Fig. 4.14 and 4.15, that the Pauli-twirled versions both of the coherent error and of the amplitude damping channels have almost the same SOF as depolarizing channels, provided that the gates used in the implementation of Pauli twirling are free from decoherence. By contrast, when Pauli twirling is implemented using realistic imperfect gates, the twirled channels have higher SOF, which is still lower than that of the amplitude damping channels. Another noteworthy fact is that imperfect Pauli twirling of two-qubit gates incurs relatively low overheads, compared to single-qubit gates. This is because two-qubit gates are more prone to decoherence than their single-qubit counterparts. Specifically, in this example we assume that every single-qubit gate (resp. two-qubit gate) has the same GGEP, and follow the convention that two-qubits gates have 10 times higher GGEP than single-qubit gates [163]. In light of these results, Pauli twirling may be a preferable quantum channel precoder, especially for two-qubit gates.

Next we illustrate the bounds of the SOF of Pauli channels presented in Section 4.2.3. As portrayed in Fig. 4.16, given a fixed GEP $\epsilon$, all points representing the SOFs of randomly produced Pauli channels fall between the upper bound (4.29) and the lower bound (4.21). Moreover, it can be seen that when the GEP is less than $5 \times 10^{-3}$, the upper and lower bounds are nearly identical, and a linear approximation of the SOF (i.e., $4\epsilon$) becomes rather accurate.

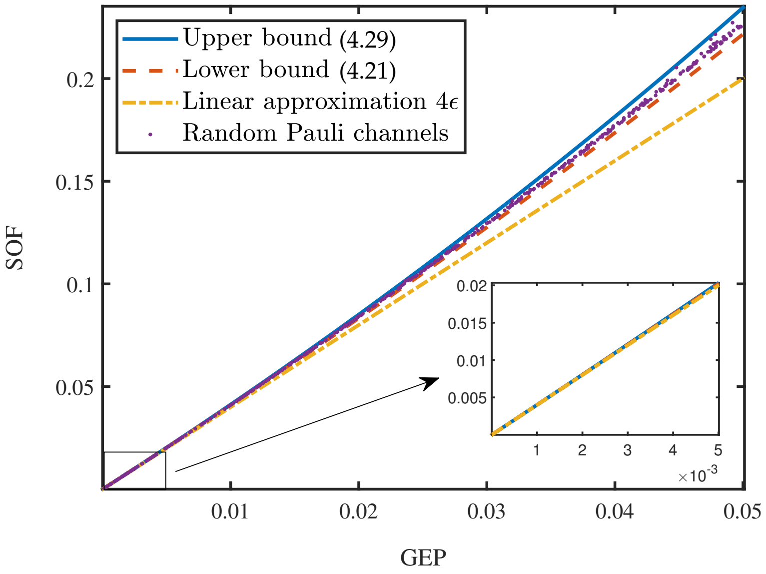


FIGURE 4.16: Upper and lower bounds of the SOF as a function of GEP $\epsilon$ for Pauli channels over a single uncoded qubit.

### 4.4.2 Transversal Gates Protected by QECC

In this subsection, we investigate the SOF when CI-QEM is applied to transversal logical gates protected by QECCs operating in their error-resilient regions. For the simplicity of presentation, we assume that all physical gates are subjected to the deleterious effect of depolarizing channels. Furthermore, we also assume that they can be decomposed into the tensor products of single-qubit depolarizing channels. Finally, we assume that all single-qubit gates have an identical GEP, and so do two-qubit gates.

As presented in Section 4.3.3, CI-QEM and QECCs operating in their error-resilient regions can be beneficially amalgamated to reduce the SOF. Here we consider the amalgam of CI-QEM and an $l$-stage concatenated Steane $[[7,1,3]]$ code. According to Proposition 4.10, for every QECC operating in its error-resilient region, there would be several critical circuit sizes. To elaborate, if a quantum circuit contains gates that exceeds the $(l+1)$-th critical point, amalgamating CI-QEM with the $l$-stage concatenated code will be more beneficial than relying on the $(l+1)$-stage concatenated code, and vice versa.

In Fig. 4.17 and 4.18, we compare the performance of three QECC-CI-QEM schemes for quantum circuits containing various number of logical gates. In Fig. 4.17, we demonstrate the aforementioned critical points of circuit size. In Fig. 4.18, areas with different shadings represent the circuit configurations for which the corresponding QECC-CI-QEM scheme is the most preferable among the three candidates. As portrayed in the figures, for the case where the GEP of physical gates equals to $10^{-4}$, pure CI-QEM

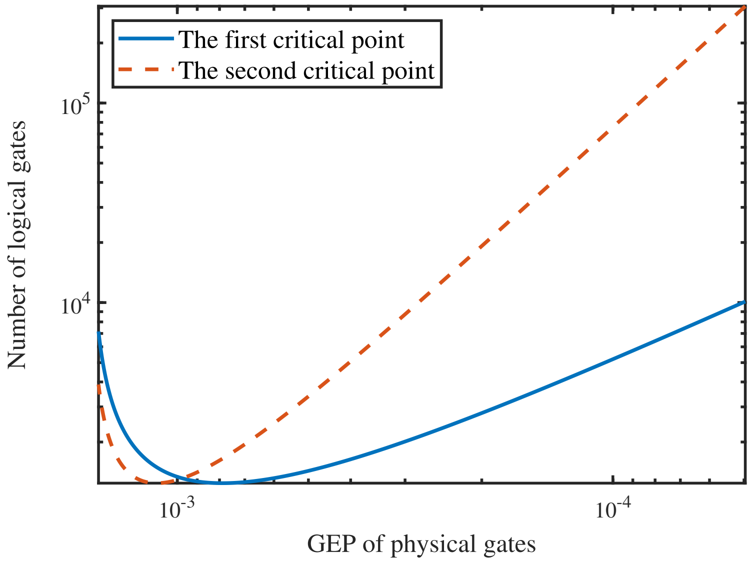


FIGURE 4.17: The critical points of circuit sizes (discussed in Proposition 4.10) versus the GEP of physical gates. Here the QECC is chosen as Steane's $[[7,1,3]]$ code.

is preferable when the circuit contains less than about $6 \times 10^3$ gates, while the single-stage QECC-CI-QEM combination may be a good choice for circuits containing between $6 \times 10^3$ and $8 \times 10^4$ gates. An interesting issue is that the pure CI-QEM becomes the most preferable when the GEP is higher than about $10^{-3}$, which is somewhat counter-intuitive. This may be attributed to the fact that the fault-tolerance threshold of the $[[7,1,3]]$ code under our assumptions in this report is around $1.5 \times 10^{-3}$. When the GEP of physical gates is close to their threshold, the error-correction capability of QECCs is not fully exploited. To elaborate a little further, the term $[f^{(l)}(\epsilon) - f^{(l+1)}(\epsilon)]$ in (4.36b) would typically be a non-monotonic function of $\epsilon$, with its maximum located close to the threshold. Hence, the critical circuit size increases as the GEP of physical gates decreases, provided that the GEP is sufficiently low.

### 4.4.3 Gates Protected by QEDC

In this subsection, we consider a combined QEDC-CI-QEM scheme, for which we make the same assumptions concerning the quantum gates as those stated in Section 4.4.2. Additionally, we assume that the GEP of two-qubit gates is 10 times as high as that of single-qubit gates.

When logical gates are implemented transversally, according to the discussion in Section 4.3.4, the total SOF of the QEDC-CI-QEM scheme would typically be even higher than that of pure CI-QEM. This is demonstrated in Fig. 4.19, where we consider the total SOF of a single transversal logical CNOT gate protected by the $[[4,2,2]]$ QEDC. It can be seen that most of the overhead is attributed to the QEDC-SOF, which is much higher than the overhead of pure CI-QEM. By contrast, the CI-QEM overhead in the

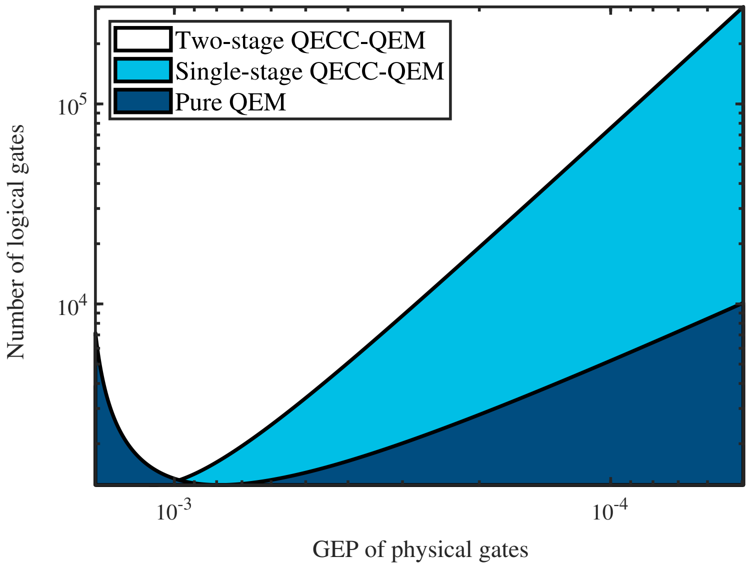


FIGURE 4.18: The most preferable QECC-CI-QEM scheme versus the GEP of physical gates. Here the QECC is chosen as Steane's $[[7,1,3]]$ code.

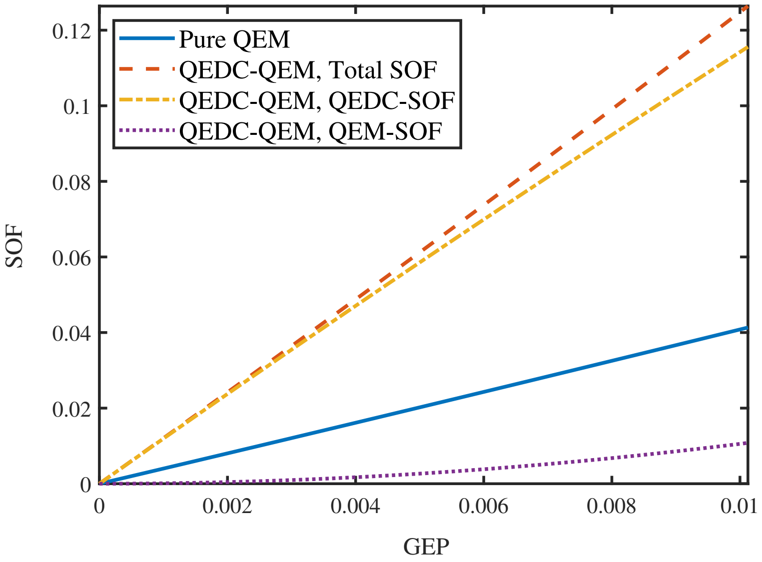


FIGURE 4.19: SOF comparison between pure CI-QEM and amalgamated $[[4,2,2]]$ QEDC-CI-QEM for a CNOT gate. The logical CNOT gates is implemented using the transversal gate configuration.

QEDC-CI-QEM scheme is significantly lower than that of pure CI-QEM, implying that the post-selection fault-tolerance threshold of the $[[4,2,2]]$ code is higher than 0.01.

As suggested by Fig. 4.11 and 4.12, some non-transversal logical gates may even outperform transversal gates in terms of requiring lower QEDC-CI-QEM SOF. In particular, we consider the specific logical gate of $\mathcal{SWAP} \circ \mathcal{H}^{\otimes 2}$ implemented in the manner illustrated in Fig. 4.12. Since this implementation only involves single-qubit gates, the

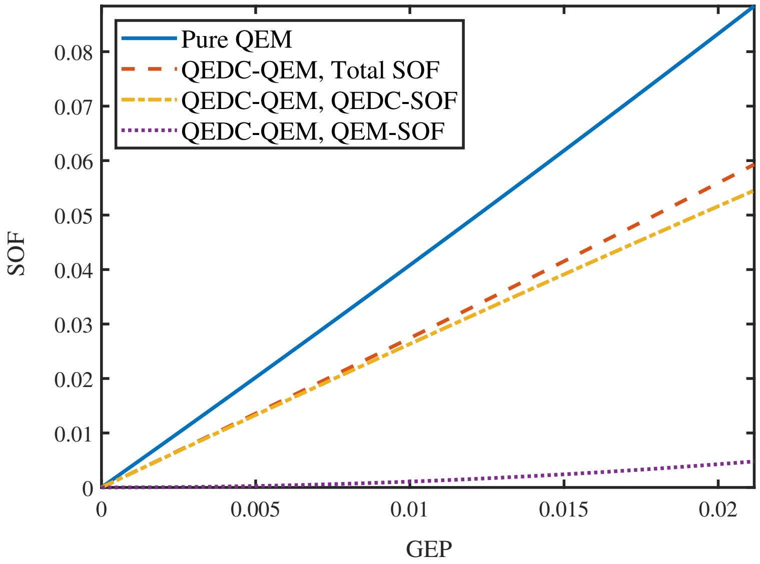


FIGURE 4.20: SOF comparison between pure CI-QEM and amalgamated [[4, 2, 2]] QEDC-CI-QEM for a $\mathcal{SWAP} \circ \mathcal{H}^{\otimes 2}$ gate. The logical $\mathcal{SWAP} \circ \mathcal{H}^{\otimes 2}$ gate is implemented as illustrated in Fig. 4.12 instead of transversally.

QEDC-SOF is significantly lower compared to the transversal implementation. Consequently, as portrayed in Fig. 4.20, the total QEDC-CI-QEM overhead is lower than that of pure CI-QEM. However, it is still not clear, whether this fact can justify the practical value of the QEDC-CI-QEM scheme, since designing a low-sampling-overhead non-transversal implementation of all Clifford gates would require substantial effort. This may be an interesting topic deserving further investigation.

## 4.5 Conclusions and Future Research

### 4.5.1 Conclusions

In this chapter, we have presented a comprehensive analysis on the SOF of CI-QEM under various channel conditions. For uncoded gates affected by errors modelled by general CPTP channels, we have shown that Pauli channels have the lowest SOF among all triangular channels (which includes the amplitude damping channels) having the same GGEP. Following this line of reasoning, we have shown furthermore that depolarizing channels have the lowest SOF in the family of all Pauli channels.

We have also conceived the QECC-CI-QEM as well as the QEDC-CI-QEM schemes, and have shown that there exist several critical quantum circuits sizes, beyond which sophisticated codes having more concatenation stages is more preferable, and vice versa.

Specifically, for QEDC-CI-QEM, we have demonstrated that it may not be compatible with the popular transversal gate configuration, but they may still have beneficial applications, when the logical gates are appropriately designed.

### 4.5.2 Future Research

- The analysis presented in this chapter relies on the assumption that the basis operations used in CI-QEM are themselves error-free. This simplifies the analysis, but may not fully reflect the practical scenario. One possible solution to this issue is to use these imperfect basis operations themselves to derive the quasi-probability representations of quantum channels, as proposed in [38]. For example, the Pauli-X gate in practice might also involve some Y-rotation and Z-rotation, but we may still use it as a basis operation, as long as this imperfection is known prior to channel inversion. Specifically, one may estimate the imperfections of basis operations using quantum gate set tomography [107], which essentially estimates the noise parameters within a relative coordinate frame. This relative coordinate frame is useful, since it is insensitive to the errors in the state preparation and measurement stages.
- Another practical issue that we did not account for in our analysis is the imperfections of the stabilizer measurements in quantum error-correction codes. Fortunately, it is a common practice to repeat the stabilizer measurements for several times when they are prone to noise [169]. Although this technique was originally proposed to enhance the error-correction performance, it also provides partial information about the quantum channels associated with these stabilizer measurements. This can be particularly useful, since other channel estimation techniques (including gate set tomography) would require additional computational overhead, which is similar in spirit to the pilot overhead in wireless communication systems. By contrast, when the repeated stabilizer measurements are exploited for channel sensing, the pilot overhead may be reduced, as seen in the code-aided solutions in wireless systems.

# Chapter 5

# Channel Inversion Based QEM: Monte Carlo Implementation

In Section 2.5.2, we have discussed a class of algorithms tailored for NISQ computers, namely that of the hybrid quantum-classical algorithms, also known as the VQAs [32, 34, 62, 170]. Briefly, VQAs aim for sharing their computational tasks between relatively simple quantum circuits and classical computers. A little more specifically, quantum circuits are employed in VQA for computing a cost function or its gradient [68], which is then fed into an optimization algorithm run on classical computers. The objective of this design paradigm is to assist near-term quantum devices in outperforming classical computers in the context of practical problems, such as solving combinatorial optimization problems using the quantum approximate optimization algorithm [32, 33, 171] and quantum chemistry problems using the variational quantum eigensolver [34].

Although the performance of VQAs has been characterized using some illustrative examples [34, 71, 175, 176], it is not fully understood whether these examples could be scaled up to problems of larger size. In fact, recent analytical results in [172, 177, 178] support the opposite statement. More explictly, [172] proves that the magnitude of the cost function (or its gradient) computed by VQAs vanishes exponentially as the number of qubits $n$ increases. Fortunately, the follow-up investigations [179, 180] found that this so-called "barren plateau" phenomenon may be mitigated to a certain extent by techniques borrowed from the literature of classical machine learning, such as pretraining and layer-by-layer training. However, the authors of [173] show that when decoherence is taken into account, the dynamic range of the computational results also vanishes exponentially upon increasing the circuit depth $N_{\mathrm{L}}$, even if these techniques are applied. To summarize, these results imply that when the quantum circuit is long in depth or large in the number of qubits, the computational error become excessive in practical applications.

TABLE 5.1: Comparison between the contributions of this chapter and existing literature evaluating the performance of VQAs and CIQEM.

| | **Circuit condition** | | **Subject of Analysis** | **Method of performance evaluation** |
|---|---|---|---|---|
| | Noisy? | QEM implementation | | |
| [62] | × | No QEM | Only accuracy | Only numerical |
| [172] | × | No QEM | | Analytical and numerical |
| [173] | ✓ | No QEM | | Analytical and numerical |
| [38] | ✓ | Exact channel inversion | Sampling overhead vs. accuracy trade-off | Only numerical |
| [1] | ✓ | Exact channel inversion | | Analytical and numerical |
| [174] | ✓ | Exact channel inversion | | Analytical and numerical |
| **This chapter** | ✓ | **Monte Carlo-based channel inversion** | | Analytical and numerical |

In Section 3.3, we have established that circuit-level QEM [37, 127, 181, 182] may be applied to improve the error scaling of VQAs with respect to the depth of the circuits. However, the error reduction capability of QEM comes at the price of a computational overhead, as discussed in Section 3.3.1.2. By appropriately choosing the total number of samples, one may strike a beneficial computational accuracy vs. overhead trade-off. Therefore, it is important to quantify the sampling overhead, before we may conclude whether QEM can play a significant role in making VQAs practical.

In this chapter, we focus our attention on the accuracy vs. overhead trade-off of CI-QEM. The literature of CI-QEM sampling overhead analysis typically assumes that the channel inversion procedure is implemented exactly [1, 38, 174]. Under this assumption, we have shown in Chapter 4 that the sampling overhead can be characterized by the sampling overhead factor (SOF), which is determined by the quality of the channel as well as by the accuracy of the basis operations implementing the channel inversion. However, perfect channel inversion may be unrealistic in practical scenarios, since it requires a pre-processing stage that is computationally excessive. Moreover, the computational cost of this pre-processing stage increases rapidly with the number of gates, which may negate the benefit of CI-QEM.

In this chapter, we consider a practical channel inversion method based on Monte Carlo sampling, which only increases the pre-processing complexity linearly with the number of gates. The drawback of this method is that it cannot invert the channel exactly, hence there would be some residual error that accumulates during computation. Compared to the ideal CI-QEM, this method has a less beneficial accuracy vs. overhead trade-off, because additional samples would be necessary to compensate for the residual error. To characterize this trade-off, we investigate the relationship between the residual error and the number of gates $N_{\mathrm{G}}$, the number of samples $N_{\mathrm{s}}$, and the gate error probability $\epsilon$. In particular, we show that the Monte Carlo based method has an error scaling on the order of $O(\sqrt{N_{\mathrm{G}}})$, while the non-QEM-protected quantum circuits exhibit an error scaling on the order of $O(N_{\mathrm{G}})$. This implies that although the Monte Carlo based method incurs residual computational errors, it still substantially outperforms the non-QEM-protected methods. We boldly and explicitly contrast our contributions to the related recent research on VQAs and CI-QEM in Table 5.1.

The rest of this chapter is organized as follows. In Section 5.1, we present the formulation of VQAs and the quantum channels modelling the decoherence. Then, in Section 5.2, we discuss a pair of CI-QEM implementation strategies, namely the Monte Carlo-based CI-QEM and the exact channel inversion. Based on this discussion, in Section 5.3, we analyse the error scaling behaviours of these two CI-QEM implementations, respectively, under the assumption that they use the same number of circuit executions. We provide further intuitions concerning these analytical results in Section 5.4, with an emphasis on the accuracy vs. sampling overhead trade-off, complemented by numerical examples in Section 5.5. Finally, we conclude in Section 5.6.

## 5.1 System Model

A typical VQA iterates between classical and quantum devices, as portrayed in Fig. 5.1. The parametric state-preparation circuit (also known as the “ansatz” [33]) transforms a fixed input state to an output state, according to the parameters chosen by a classical optimizer. The output state is then measured and fed into a quantum observable, which maps the measurement outcomes to the desired computational results. The results correspond to the value of a cost function or its gradient, which in turn serve as the input of the associated classical optimization algorithm. The iterations continue until certain stopping criterion is met, for example, the computed gradient becomes almost zero.

In this chapter, we focus on the error induced by the sampling procedure in CI-QEM, hence we consider the computational result of a single iteration, meaning that the parameters used for state preparation are fixed. We model the decoherence-induced impairment in the parametric state-preparation circuit as quantum channels acting upon the associated quantum states at the output of perfect quantum gates, as exemplified by the simple circuit shown in Fig. 5.2. In this figure, $\mathcal{C}_k,\ k = 1 \ldots 4$ represents the channel modelling the decoherence in the $k$-th quantum gate, while $\mathcal{G}_k$ represents the $k$-th ideal decoherence-free quantum gate.

In the subsequent subsections, we present the mathematical formulations of the system models shown in Fig. 5.1 and 5.2.

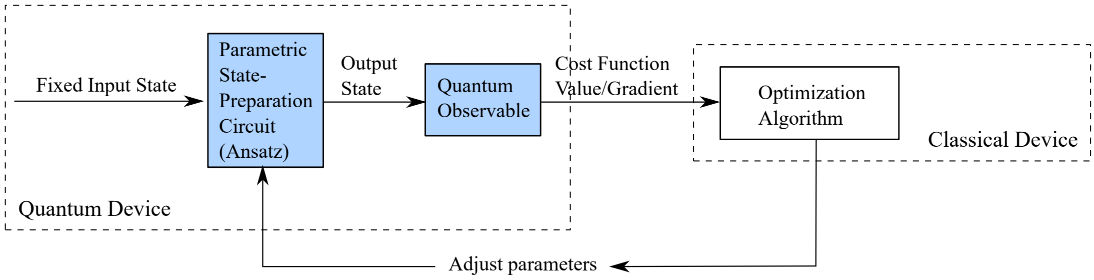


FIGURE 5.1: The structure of a typical implementation for variational quantum algorithms.

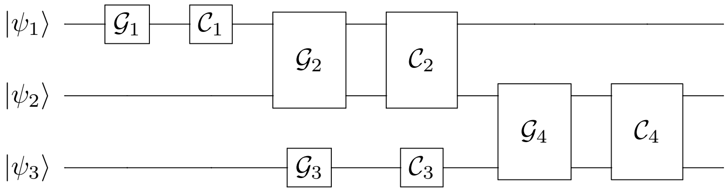


FIGURE 5.2: Simple example of the noisy parametric state-preparation circuit seen in Fig. 5.1.

### 5.1.1 Operator-sum Representation

Without loss of generality, we assume that the input state of the circuit is the all-zero state $|0\rangle^{\otimes n}$, where $n$ is the number of qubits. In general, when the circuit is decoherence-free, the computational result of a variational quantum circuit may be expressed as

$$\widetilde{r} = \langle 0|^{\otimes n} \left( \prod_{k=1}^{N_\mathrm{G}} \mathcal{G}_i^\dagger \right) \mathcal{M}_\mathrm{ob} \left( \prod_{k=1}^{N_\mathrm{G}} \mathcal{G}_{N_\mathrm{G}-k+1} \right) |0\rangle^{\otimes n}, \tag{5.1}$$

where $N_\mathrm{G}$ is the number of gates in the circuit, $\mathcal{G}_k$ denotes the $k$-th quantum gate, and the operator $\mathcal{M}_\mathrm{ob}$ represents the quantum observable, which describes the computational task as a linear function of the final state.

If we consider a more practical scenario, where the quantum state evolves owing to quantum decoherence as the circuit operates, the state can no longer be fully characterized using the state vector formalism. Instead, we may use the density matrix formalism. In particular, the input state may be described as

$$\rho_0 = (|0\rangle \langle 0|)^{\otimes n}. \tag{5.2}$$

Correspondingly, the output state of the $k$-th imperfect quantum gate may be represented in an operator-sum form [48, Sec. 8.2.4], relying on following recursive relationship

$$\rho_k = \mathcal{C}_k \left( \mathcal{G}_k \rho_{k-1} \mathcal{G}_k^\dagger \right), \tag{5.3}$$

where the operator $\mathcal{C}_k$ is characterized by

$$\mathcal{C}_k(\rho) = \sum_{i=1}^{n_k} \left( \boldsymbol{E}_{k,i} \rho \boldsymbol{E}_{k,i}^{\dagger} \right), \tag{5.4}$$

representing the channel modelling the imperfection of the $k$-th gate. The matrices $\boldsymbol{E}_{k,i}$ represent the operation elements [48, Sec. 8.2.4] of the channel $\mathcal{C}_k$ satisfying the completeness condition of $\sum_{i=1}^{n_k} \boldsymbol{E}_{k,i}^{\dagger} \boldsymbol{E}_{k,i} = \boldsymbol{I}$. Finally, when all gates completed their tasks and the measurement results have been obtained, the computational result may be expressed as

$$r = \mathrm{Tr}\left\{ \mathcal{M}_{\mathrm{ob}} \rho_{N_{\mathrm{G}}} \right\}. \tag{5.5}$$

### 5.1.2 Pauli Transfer Matrix Representation

In the standard operator-sum form [48, Sec. 8.2.4], the quantum states are represented by matrices. However, in many applications, such as the error analysis considered in this treatise, it would be more convenient to treat them as vectors. Correspondingly, the quantum channels and gates would then be represented by matrices. To this end, the PTM representation of quantum operators was proposed in [183], which allows a quantum operator $\mathcal{O}$ to be expressed as

$$[\boldsymbol{O}]_{i,j} = \frac{1}{2^n} \mathrm{Tr}\left\{ \mathcal{S}_i \mathcal{O}(\mathcal{S}_j) \right\}, \tag{5.6}$$

where $\mathcal{S}_i$ denotes the $i$-th Pauli operator in the $n$-qubit Pauli group. Similarly, a quantum state $\rho$ can be expressed as

$$[\boldsymbol{\rho}]_i = \frac{1}{\sqrt{2^n}} \mathrm{Tr}\left\{ \mathcal{S}_i \rho \right\}. \tag{5.7}$$

Under the PTM representation, the computational result may be rewritten as

$$r = \boldsymbol{v}_{\mathrm{ob}}^{\mathrm{T}} \left( \prod_{k=1}^{N_{\mathrm{G}}} (\boldsymbol{C}_{N_{\mathrm{G}}-k+1} \boldsymbol{G}_{N_{\mathrm{G}}-k+1}) \right) \boldsymbol{v}_0, \tag{5.8}$$

where $\boldsymbol{G}_k$ represents the $k$-th perfect gate, and $\boldsymbol{C}_k$ represents the channel modelling the imperfection of the $k$-th gate. The vector $\boldsymbol{v}_0$ denotes the initial state, whereas $\boldsymbol{v}_{\mathrm{ob}}$ is the vector representation of the quantum observable $\mathcal{M}_{\mathrm{ob}}$.

To simplify the notation, we define

$$\begin{aligned} \boldsymbol{R}_k &:= \prod_{i=1}^{k} (\boldsymbol{C}_{k-i+1} \boldsymbol{G}_{k-i+1}), \\ \widetilde{\boldsymbol{R}}_k &:= \prod_{i=1}^{k} \boldsymbol{G}_{k-i+1}. \end{aligned} \tag{5.9}$$

Especially, for $k = 0$, we define $\boldsymbol{R}_0 = \widetilde{\boldsymbol{R}}_0 = \boldsymbol{I}$. The output state of the $k$-th quantum gate can then be expressed as

$$\begin{aligned} \boldsymbol{v}_k &:= \boldsymbol{R}_k \boldsymbol{v}_0 \\ &= \boldsymbol{C}_k \boldsymbol{G}_k \boldsymbol{v}_{k-1}. \end{aligned} \tag{5.10}$$

Hence we have

$$\begin{aligned} r &= \boldsymbol{v}_{\mathrm{ob}}^{\mathrm{T}} \boldsymbol{v}_{N_{\mathrm{G}}} \\ &= \boldsymbol{v}_{\mathrm{ob}}^{\mathrm{T}} \boldsymbol{R}_{N_{\mathrm{G}}} \boldsymbol{v}_0. \end{aligned} \tag{5.11}$$

### 5.1.3 Channel Model

In this treatise, we consider Pauli channels [184], for which the Pauli transfer matrices take the following form

$$\boldsymbol{C}_k = \mathrm{diag}\left\{\boldsymbol{c}_k\right\}, \tag{5.12}$$

where

$$\boldsymbol{c}_k = \widetilde{\boldsymbol{H}} \boldsymbol{p}_k, \tag{5.13}$$

with $\widetilde{\boldsymbol{H}}$ denoting the Hadamard transform, whereas $\boldsymbol{p}_k$ represents a probability distribution satisfying $\mathbf{1}^{\mathrm{T}} \boldsymbol{p}_k = 1$, $\boldsymbol{p}_k \geq 0$.

## 5.2 CI-QEM and Its Implementation Strategies

Ideally, for a channel $\boldsymbol{C}_k$, CI-QEM would apply its inverse based on a linear combination of predefined quantum operations, taking the following form

$$\boldsymbol{C}_k^{-1} = \sum_{l=1}^{L} \alpha_l^{(k)} \boldsymbol{O}_l, \tag{5.14}$$

where $\boldsymbol{O}_l$ is the $l$-th quantum operation, while $\boldsymbol{\alpha}_k := [\alpha_1 \ \ldots \ \alpha_L]^{\mathrm{T}}$ is the quasi-probability representation vector satisfying $\mathbf{1}^{\mathrm{T}} \boldsymbol{\alpha}_k = 1$. This linear combination may be rewritten as a probabilistic mixture of the quantum operations as follows:

$$\boldsymbol{C}_k^{-1} = \|\boldsymbol{\alpha}_k\|_1 \sum_{l=1}^{L} s_l^{(k)} p_l^{(k)} \boldsymbol{O}_l, \tag{5.15}$$

where $s_l^{(k)}$ and $p_l^{(k)}$ are the $l$-th entries of $\boldsymbol{s}_k$ and $\boldsymbol{p}_k$, respectively, given by

$$\begin{aligned} p_i^{(k)} &= \frac{|\alpha_i^{(k)}|}{\|\boldsymbol{\alpha}_k\|_1}, \\ \boldsymbol{s}_k &= \mathrm{sgn}\{\boldsymbol{\alpha}_k\}. \end{aligned} \tag{5.16}$$

Note that the vector $\boldsymbol{p}_k$ describes a probability distribution.

Typically, the probabilistic mixture in (5.15) is implemented by generating a set of candidate circuits and performing post-processing on the output of these circuits. In the following subsections, we will discuss different candidate selection strategies and their characteristics.

### 5.2.1 Exact Implementation and Sampling Overhead

The inverse channel $\boldsymbol{C}_k^{-1}$ in (5.15) is assumed to be implemented exactly in the seminal paper [37] that proposed CI-QEM for the first time, as well as in many other existing contributions [1,38,174]. Exact implementation implies that, each quantum operation $\boldsymbol{O}_l$ should appear in exactly $Np_l^{(k)}$ candidate circuits in every $N$ samples of the computational result.

The assumption of exact implementation significantly simplifies the performance analysis of QEM. In particular, it leads to a clear and concise formula of sampling overhead, which describes the computational overhead imposed by the variance-boosting effect of QEM. To elaborate, assume that the variance of the computational result is $\sigma^2$ based on $N_0$ samples. According to (5.15), if we implement the inverse channel $\boldsymbol{C}_k^{-1}$, the variance would become $\|\boldsymbol{\alpha}_k\|_1^2\sigma^2$. Therefore, in order to achieve the same accuracy as the case without QEM, we should acquire $N_0(\|\boldsymbol{\alpha}_k\|_1^2 - 1)$ additional samples. If we further assume that all gates are protected by CI-QEM, we have the following formula for the total sampling overhead

$$N_{\text{exact}} = N_0 \left( \prod_{k=1}^{N_{\text{G}}} \|\boldsymbol{\alpha}_k\|_1^2 - 1 \right). \tag{5.17}$$

The simplicity of (5.17) is largely due to the assumption of exact implementation.

Despite its theoretical convenience, the practicality of exact implementation is doubtful. Specifically, the number of the $l$-th candidate circuit, $Np_l^{(k)}$ has to be an integer, which might be unrealistic for an arbitrary $p_l^{(k)}$. Furthermore, the number of the probability parameters $p_l^{(k)}$ would increase exponentially as the number of CI-QEM-protected gates increases, which may render the candidate circuit selection procedure computationally prohibitive when $N_{\text{G}}$ is large. Motivated by these drawbacks, we consider using a Monte Carlo implementation of CI-QEM, detailed in the next subsection.

### 5.2.2 Monte Carlo Implementation

In the Monte Carlo implementation, we first sample from the probability distribution $\boldsymbol{p}_k$ for each gate, and obtain $N$ samples constituting a set $\mathsf{L} = \{\mathsf{l}_1, \ldots, \mathsf{l}_N\}$, where for all

$k$ we have $\mathsf{l}_k = 1, 2, \ldots, L$. Thus we may approximate the inverse channel as

$$\begin{aligned} \boldsymbol{\Gamma}_k &= \frac{\|\boldsymbol{\alpha}_k\|_1}{N} \sum_{i=1}^{N} s_{\mathsf{l}_i}^{(k)} \boldsymbol{O}_{\mathsf{l}_i} \\ &= \|\boldsymbol{\alpha}_k\|_1 \sum_{l=1}^{L} s_l^{(k)} \widetilde{\mathsf{p}}_l^{(k)} \boldsymbol{O}_l, \end{aligned} \tag{5.18}$$

where

$$\widetilde{\mathsf{p}}_m^{(k)} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\{\mathsf{l}_i = m\}.$$

The advantage of the Monte Carlo approach is that it may result in a much lower complexity for candidate circuit generation, compared to the exact implementation. To elaborate further, as a "toy" example, for a circuit consisting of two gates we have

$$\boldsymbol{\Gamma}_2 \widetilde{\boldsymbol{G}}_2 \boldsymbol{\Gamma}_1 \widetilde{\boldsymbol{G}}_1 = \frac{\|\boldsymbol{\alpha}_1\|_1 \|\boldsymbol{\alpha}_2\|_1}{N} \sum_{i=1}^{N} s_{\mathsf{l}_{i,1}}^{(1)} s_{\mathsf{l}_{i,2}}^{(2)} \boldsymbol{O}_{\mathsf{l}_{i,2}} \widetilde{\boldsymbol{G}}_2 \boldsymbol{O}_{\mathsf{l}_{i,1}} \widetilde{\boldsymbol{G}}_1, \tag{5.19}$$

where $\widetilde{\boldsymbol{G}}_k = \boldsymbol{C}_k \boldsymbol{G}_k$, and $\mathsf{l}_{i,k}$ denotes the $i$-th sample drawn from the distribution $\boldsymbol{p}_k$. This implies that in order to obtain a sample for the entire circuit, we may simply generate one sample for each gate, and concatenate them as shown in the right hand side of (5.19). Compared to the exact implementation, the Monte Carlo implementation can generate an arbitrary number of circuit samples $N$, at a relatively low computational cost of $O(NN_{\mathrm{G}})$.

The reduced complexity of the Monte Carlo implementation comes with a cost of inaccurate channel inversion, since $\boldsymbol{G}_k$ is only an approximation of $\boldsymbol{C}_k^{-1}$. Hence there would be a residual channel for each gate, which is given by

$$\widetilde{\mathbf{C}}_k = \boldsymbol{\Gamma}_k \boldsymbol{C}_k. \tag{5.20}$$

A natural question that arises is, whether the additional computational error caused by these residual channels would erode the error reduction capability of Monte Carlo-based QEM. In the rest of this treatise, we will discuss the impact of these residual channels on the accuracy vs. sampling overhead trade-off.

## 5.3 Error Scaling Analysis of Monte Carlo Based CI-QEM

In this section, we discuss the error scaling behaviour of quantum circuits protected by Monte Carlo-based CI-QEM, and contrast the results to that of circuits without QEM protection. In order to make a fair comparison, we consider the following assumptions.

### 5.3.1 Assumptions

*Assumption* 2 (Bounded gate error rate). The error probability of each quantum gate is upper bounded by $\epsilon_{\mathrm{u}}$.

Since we consider Pauli channels in this treatise, the gate error probability corresponding to a quantum channel $\boldsymbol{C}_k$ (under its PTM representation) may be computed as

$$\epsilon(\boldsymbol{C}_k) = 1 - \frac{1}{4^n}\mathrm{Tr}\left\{\boldsymbol{C}_k\right\}. \tag{5.21}$$

*Assumption* 3 (Bounded observable). The eigenvalues of the quantum observable $\mathcal{M}_{\mathrm{ob}}$ are bounded in the interval $[-1, 1]$.

Assumption 3 ensures the boundedness of the computation result $r$. In this treatise we assume that the upper and lower bounds are 1 and $-1$, respectively, but they may be replaced with any other constant real numbers without affecting our analytical results. The assumption may also be rewritten as

$$\max_{\boldsymbol{v}\in\mathbb{S}^n} \boldsymbol{v}_{\mathrm{ob}}^{\mathrm{T}}\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}_{\mathrm{ob}} \leq 1, \tag{5.22}$$

where $\mathbb{S}^n$ denotes the space of all density matrices over $n$ qubits. Furthermore, the assumption also implies that

$$\|\boldsymbol{v}_{\mathrm{ob}}\|_2 \leq \sqrt{2^n}. \tag{5.23}$$

This follows from the fact that $\|\boldsymbol{v}_{\mathrm{ob}}\|_2 = \|\mathcal{M}_{\mathrm{ob}}\|_{\mathrm{F}}$, and that

$$\begin{aligned}\|\mathcal{M}_{\mathrm{ob}}\|_{\mathrm{F}} &= \sum_{i=1}^{2^n} \lambda_i(\mathcal{M}_{\mathrm{ob}}) \\ &\leq \sqrt{2^n},\end{aligned}$$

where $\lambda_i(\cdot)$ denotes the $i$-th largest eigenvalue of its argument.

*Assumption* 4 (Zero bias term). We assume that

$$\mathrm{Tr}\left\{\mathcal{M}_{\mathrm{ob}}\right\} = \sqrt{2^n}[\boldsymbol{v}_{\mathrm{ob}}]_1 = 0. \tag{5.24}$$

Note that $[\boldsymbol{v}_{\mathrm{ob}}]_1$ is the coefficient of the identity operator, which serves as a bias term in the computation result being constant with respect to the quantum state. Thus this assumption does not restrict the generality of our results.

### 5.3.2 Benchmark: Error Scaling in the Absence of QEM

In this subsections, we characterize the error scaling of quantum circuits that are not protected by QEM. The results will serve as important benchmarks in the following discussions. Let us start with a bound of the dynamic range of computational results, which will lead to a lower bound of the computational error.

**Proposition 5.1.** *Assume that each qubit would be processed by at least $N_{\mathrm{L}}$ gates, and that for each of these gates, the probability of each type of Pauli error (i.e., X error, Y error or Z error) on each qubit is lower bounded by $\epsilon_{\mathrm{l}}$. The computational result $r$ exhibits the following convergence behaviour:*

$$|r| \leq \exp\left(-4\epsilon_{\mathrm{l}} N_{\mathrm{L}}\right). \tag{5.25}$$

*Proof.* Please refer to Appendix A.5. □

Proposition 5.1 implies that decoherence would force the computation result to be almost independent of the quantum observable $\boldsymbol{v}_{\mathrm{ob}}$ in an asymptotic sense. Indeed, as indicated by (5.25), when $N_{\mathrm{L}}$ is large, $r$ is only determined by the first entry of $\boldsymbol{v}_{\mathrm{ob}}$. Moreover, consider the case where $|\widetilde{r}| \geq 1-c$ holds for all $N_{\mathrm{G}}$, the computational error is lower bounded as

$$|r-\widetilde{r}| \geq 1-c-\exp\left(-4\epsilon_{\mathrm{l}} N_{\mathrm{L}}\right). \tag{5.26}$$

From the Taylor expansion

$$\exp\left(-4\epsilon_{\mathrm{l}} N_{\mathrm{L}}\right) = 1-4\epsilon_{\mathrm{l}} N_{\mathrm{L}} + \frac{(4\epsilon_{\mathrm{l}} N_{\mathrm{L}})^2}{2} - \cdots,$$

we see that when $\epsilon_{\mathrm{L}} N_{\mathrm{L}} \ll 1$, the lower bound is approximately

$$|r-\widetilde{r}| \gtrapprox 4\epsilon_{\mathrm{l}} N_{\mathrm{L}} - c, \tag{5.27}$$

which increases linearly with respect to $\epsilon_{\mathrm{l}} N_{\mathrm{L}}$.

We may also provide an upper bound for the computational error as follows.

**Proposition 5.2.** *The computational error can be upper bounded as*

$$|r-\widetilde{r}| \leq 2\epsilon_{\mathrm{u}} N_{\mathrm{G}}. \tag{5.28}$$

*Proof.* Please refer to Appendix A.6. □

Combining Propositions 5.1 and 5.2, we see that the computational error grows linearly with $N_{\mathrm{G}}$, when the number of gates in each "layer" is constant (hence $N_{\mathrm{L}}$ is a constant multiple of $N_{\mathrm{G}}$. This is typically true for VQAs.

### 5.3.3 The Statistics of the Residual Channels

Before diving into details about the error scaling, in this subsection, we first investigate the characteristics of the residual channels of gates protected by Monte Carlo-based CI-QEM.

According to the sampling overhead analysis in [1] based on the assumption of exact channel inversion, if we wish to execute the decoherence-free circuit $N_{\mathrm{s}}$ times, we should sample from the probabilistic mixture of candidate circuits for as many as

$$N = N_{\mathrm{s}} \|\boldsymbol{\alpha}_k\|_1^2 \tag{5.29}$$

times, in order to keep the variance of the computational result unchanged by the channel inversion procedure. Here we consider the Monte Carlo-based channel inversion using the same number of samples, hence we have

$$\widetilde{\mathsf{p}}_m^{(k)} = \frac{1}{N_s \|\boldsymbol{\alpha}_k\|_1^2} \sum_{i=1}^{N_s \|\boldsymbol{\alpha}_k\|_1^2} \mathbb{I}\{\mathsf{I}_i = m\}. \tag{5.30}$$

Of course, the Monte Carlo-based channel inversion has lower accuracy compared to the exact channel inversion, when they use the same number of samples. The accuracy could be improved by using additional samples, which will be discussed in more detail in Section 5.4.2.

After the sampling procedure, $\boldsymbol{\alpha}_k$ is approximated by $\widetilde{\boldsymbol{\alpha}}_k$ taking the following form

$$\begin{aligned}
\widetilde{\boldsymbol{\alpha}}_k &= \|\boldsymbol{\alpha}_k\|_1 \cdot \boldsymbol{s}_k \odot \widetilde{\mathbf{p}}_k \\
&= \|\boldsymbol{\alpha}_k\|_1 \cdot \boldsymbol{s}_k \odot (\boldsymbol{p}_k + \mathbf{n}) \\
&= \boldsymbol{\alpha}_k + \|\boldsymbol{\alpha}_k\|_1 \cdot \boldsymbol{s}_k \odot \mathbf{n},
\end{aligned} \tag{5.31}$$

where $\mathbf{n}$ denotes the sampling error. In general, the approximated inverse channel may be expressed in terms of $\widetilde{\boldsymbol{\alpha}}_k$ as

$$\boldsymbol{\Gamma}_k = \sum_{i=1}^{L} [\widetilde{\boldsymbol{\alpha}}_k]_i \boldsymbol{B}_i,$$

where $\{\boldsymbol{B}_i\}_{i=1}^{L}$ is a set of operators forming a basis of the space where the imperfect gate $\boldsymbol{C}_k \boldsymbol{G}_k$ resides in. Interested readers may refer to Table 1 in [1] for an example of such operator sets. For the Pauli channels considered in this treatise, $\boldsymbol{\Gamma}$ has a simpler form. Specifically, using (5.12), the quasi-probability representation vector may be expressed as

$$\boldsymbol{\alpha}_k = \widetilde{\boldsymbol{H}}^{-1}(1/\boldsymbol{c}_k), \tag{5.32}$$

where $\widetilde{\boldsymbol{H}}$ is the Hadamard transform over $n$ qubits, and $\widetilde{\boldsymbol{H}}^{-1}$ is the corresponding inverse transform given by $\widetilde{\boldsymbol{H}}^{-1} = \frac{1}{4^n}\widetilde{\boldsymbol{H}}$. The approximated inverse channel can now be expressed as

$$\mathsf{\Gamma}_k = \mathrm{diag}^{-1}\{\widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{\alpha}}_k\}, \tag{5.33}$$

and thus the residual channel takes the following form

$$\widetilde{\mathsf{C}}_k = \mathrm{diag}^{-1}\{\widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{\alpha}}_k \odot \boldsymbol{c}_k\}. \tag{5.34}$$

To simplify our further analysis, we introduce $\widetilde{\mathbf{c}}_k := \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{\alpha}}_k \odot \boldsymbol{c}_k$, where $\widetilde{\mathbf{c}}_k$ may be further expressed as

$$\begin{aligned} \widetilde{\mathbf{c}}_k &= \mathbf{1} + \|\boldsymbol{\alpha}_k\|_1 \cdot \widetilde{\boldsymbol{H}}(\boldsymbol{s}_k \odot \mathbf{n}) \odot \boldsymbol{c}_k \\ &= \mathbf{1} + \|\boldsymbol{\alpha}_k\|_1 \cdot \boldsymbol{c}_k \odot \widetilde{\mathbf{n}}, \end{aligned} \tag{5.35}$$

and $\widetilde{\mathbf{n}} := \widetilde{\boldsymbol{H}}(\boldsymbol{s}_k \odot \mathbf{n})$. Note that the vector $\widetilde{\mathbf{p}}_k$ is a multinomial distributed random vector, satisfying

$$\begin{aligned} \mathbb{E}\{\widetilde{\mathbf{p}}_k\} &= \boldsymbol{p}_k, \\ \mathrm{Cov}\{\widetilde{\mathbf{p}}_k\} &= \frac{1}{N_\mathrm{s}\|\boldsymbol{\alpha}_k\|_1^2}\left(\boldsymbol{P}_k - \boldsymbol{p}_k\boldsymbol{p}_k^\mathrm{T}\right), \end{aligned} \tag{5.36}$$

where $\boldsymbol{P}_k = \mathrm{diag}\,\{\boldsymbol{p}_k\}$. Therefore, the vector $\widetilde{\mathbf{n}}$ satisfies

$$\begin{aligned} \mathbb{E}\{\widetilde{\mathbf{n}}\} &= \mathbf{0}, \\ \mathrm{Cov}\{\widetilde{\mathbf{n}}\} &= \widetilde{\boldsymbol{H}}\mathrm{Cov}\{\widetilde{\mathbf{p}}_k\}\widetilde{\boldsymbol{H}}, \end{aligned} \tag{5.37}$$

since the sign vector $\boldsymbol{s}_k$ does not have an effect on the covariance matrix. Thus we have the following results for $\widetilde{\mathbf{c}}_k$:

$$\begin{aligned} \mathbb{E}\{\widetilde{\mathbf{c}}_k\} &= \mathbf{1}, \\ \mathrm{Cov}\{\widetilde{\mathbf{c}}_k\} &= \|\boldsymbol{\alpha}_k\|_1^2 \cdot \mathrm{Cov}\{\boldsymbol{c}_k \odot \widetilde{\mathbf{n}}\} \\ &= \frac{1}{N_\mathrm{s}}\widetilde{\boldsymbol{H}}\left(\boldsymbol{P}_k - \boldsymbol{p}_k\boldsymbol{p}_k^\mathrm{T}\right)\widetilde{\boldsymbol{H}} \odot \boldsymbol{c}_k\boldsymbol{c}_k^\mathrm{T}. \end{aligned} \tag{5.38}$$

For simplicity of further derivation, we use the notation of $\boldsymbol{\Xi}_k := \mathrm{Cov}\{\widetilde{\mathbf{c}}_k\}$.

### 5.3.4 Error Scaling in the Presence of Monte Carlo Based CI-QEM

In this subsection, we investigate the scaling law of computational error when the quantum circuit is protected by Monte Carlo-based CI-QEM, based on the above discussions concerning the residual channels in the previous subsection.

We note that for CI-QEM-protected circuits, the computational result is a random variable due to the randomness in the sampling procedure, given by

$$\mathsf{r} = \boldsymbol{v}_\mathrm{ob}^\mathrm{T}\mathbf{v}_{N_\mathrm{G}}, \tag{5.39}$$

where

$$\mathsf{v}_k = \mathsf{R}_k v_0 = \widetilde{\mathsf{C}}_k G_k \mathsf{v}_{k-1}. \tag{5.40}$$

Using these definitions, we have the following bound on the RMSE of the computational result $\mathsf{r}$.

**Proposition 5.3** (Square-root Increasing QEM Inaccuracy). *For a quantum circuit consisting of $N_{\mathrm{G}}$ gates which is protected by QEM, the RMSE of the computational result is upper bounded by*

$$\sqrt{\mathbb{E}\{(\mathsf{r}-\widetilde{r})^2\}} \leq 2^{n/2}\sqrt{\exp(2N_{\mathrm{G}}N_{\mathrm{s}}^{-1})-1}. \tag{5.41}$$

*Proof.* Please refer to Appendix A.7. □

Note that by applying the Taylor expansion to $\exp(2N_{\mathrm{G}}N_{\mathrm{s}})$, we have

$$\exp(2N_{\mathrm{G}}N_{\mathrm{s}})-1 = \frac{2}{N_{\mathrm{s}}}N_{\mathrm{G}} + \frac{1}{2}\left(\frac{2}{N_{\mathrm{s}}}N_{\mathrm{G}}\right)^2 + \cdots,$$

which is approximately $2N_{\mathrm{G}}N_{\mathrm{s}}^{-1}$, when $N_{\mathrm{G}} \ll N_{\mathrm{s}}$. This means that when the RMSE is far less than 1, its scaling law is given by $O(\sqrt{N_{\mathrm{G}}}/\sqrt{N_{\mathrm{s}}})$. This is particularly useful, since in typical applications (e.g., variational quantum algorithms), having an RMSE close to 1 would be excessive.

In Proposition 5.3, the dependence of the RMSE on the error probability of quantum gates is not demonstrated. According to (A.67) of the Appendix, this dependence mainly relies on the term $\|\Xi_k\|_{\max}$. Next we expound a little further on this issue based on Assumption 2.

**Proposition 5.4.** *Under Assumption 2, we have the following refined upper-bound for the RMSE of the computational result:*

$$\begin{aligned}\sqrt{\mathbb{E}\{(\mathsf{r}-\widetilde{r})^2\}} &\leq 2^{n/2}\sqrt{\exp\left(\tilde{\epsilon}N_{\mathrm{G}}N_{\mathrm{s}}^{-1}\right)-1} \\ &\approx 2^{n/2}\sqrt{\exp\left(10\epsilon_{\mathrm{u}}N_{\mathrm{G}}N_{\mathrm{s}}^{-1}\right)-1}\end{aligned} \tag{5.42}$$

*where $\tilde{\epsilon}$ is given by*

$$\tilde{\epsilon} := \frac{5}{2}\sigma_{\mathrm{u}} + \frac{1}{4}\sigma_{\mathrm{u}}^2, \tag{5.43}$$

*and*

$$\sigma_{\mathrm{u}} := 4\epsilon_{\mathrm{u}} \cdot \frac{1-\epsilon_{\mathrm{u}}}{(1-2\epsilon_{\mathrm{u}})^2}. \tag{5.44}$$

*The approximation is valid when $\epsilon_{\mathrm{u}} \ll 1$.*

*Proof.* Please refer to Appendix A.8. □

Proposition 5.4 implies that, when $\epsilon_{\mathrm{u}} N_{\mathrm{G}} \ll N_{\mathrm{s}}$, the RMSE is on the order of $O(\sqrt{\epsilon_{\mathrm{u}} N_{\mathrm{G}}}/\sqrt{N_{\mathrm{s}}})$.

*Remark* 5.5. We note that the factor $2^{n/2}$ in (5.41) and (5.42) is an artifact of our specific proof technique. According to the numerical results which will be presented in Section 5.5, we conjecture this scaling factor is essentially unnecessary, implying that

$$\sqrt{\mathbb{E}\{(\mathrm{r}-\widetilde{r})^2\}} \leq \sqrt{\exp\left(\epsilon N_{\mathrm{G}} N_{\mathrm{s}}^{-1}\right) - 1}. \tag{5.45}$$

However, it seems to be technically challenging to remove this factor from the bounds. Further investigations into this issue will be left for our future research.

## 5.4 Discussions

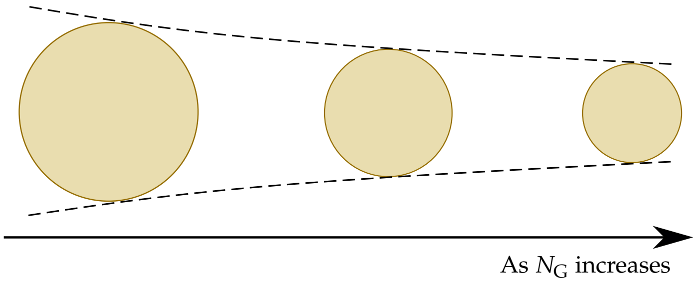
In this section, we provide further intuitions concerning the analytical results of Section 5.3.

### 5.4.1 Intuitions about the Error Scaling with the Circuit Size

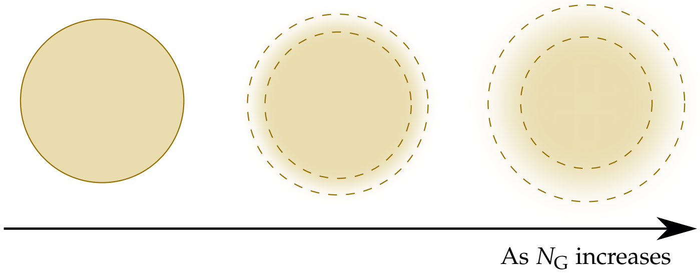
The results in Section 5.3 indicate that, with respect to $N_{\mathrm{G}}$, we observe an $O(\sqrt{N_{\mathrm{G}}})$ scaling of the computational error of circuits protected by Monte Carlo-based CI-QEM, when the number of samples is the same as that of CI-QEM based on exact channel inversion. By contrast, when QEM is not applied, the computational error scales as $O(N_{\mathrm{G}})$, as discussed in Section 5.3. Thus we may conclude that, although there are residual channels due to the inexact channel inversion, Monte Carlo-based CI-QEM can still slow down the accumulation of computational error.

We may understand these error scaling behaviours more intuitively by considering the low-complexity example of a single-qubit circuit, where the entire space of all legitimate quantum states can be described by the celebrated Bloch sphere [48, Sec. 1.2] As demonstrated in Fig. 5.3, the Bloch sphere would shrink as $N_{\mathrm{G}}$ increases when no QEM is applied, since the Pauli channels are contractive. This is in stark contrast with the case where Monte Carlo-based CI-QEM is applied, when the Bloch sphere becomes "blurred" as $N_{\mathrm{G}}$ increases, since it is not determined whether the sphere will expand or shrink after each gate. Consequently, the sphere may expand after one gate and then shrink after another, hence the corresponding errors will cancel each other to certain extent. In light of the aforementioned intuition, we may interpret the error scaling of Monte Carlo-based CI-QEM in following informal way.

Assume that every gate $k$ would transform the Bloch sphere in a way that its radius becomes $(1 + \lambda_k)$ times that of its original value, where $\lambda_k$ is a zero-mean random variable with variance $\sigma_k^2$. If additionally all $\lambda_k$ values are mutually independent, we can



(a) Gates without QEM protection



(b) Gates protected by QEM

FIGURE 5.3: Graphical illustration of the Bloch sphere undergoing a sequence of $N_{\mathrm{G}}$ imperfect gates. The Bloch sphere shrinks when QEM is not applied, whereas it becomes "blurred" when QEM is applied.

see that

$$\frac{1}{N_{\mathrm{G}}}\sum_{k=1}^{N_{\mathrm{G}}}\ln(1+\lambda_k)\sim\mathcal{N}\left(-\frac{1}{2}\sigma^2,\sigma^2\right)$$

holds asymptotically as $N_{\mathrm{G}}\to\infty$ by applying the central limit theorem, where we have:

$$\sigma^2=\frac{1}{N_{\mathrm{G}}}\sum_{k=1}^{N_{\mathrm{G}}}\sigma_k^2.$$

Hence the radius of the Bloch sphere after $k$ gates, denoted by $\mathsf{a}_k$, tends to be a log-normally distributed random variable characterized by

$$\mathbb{E}\{\mathsf{a}_k\}=1,$$
$$\mathrm{Var}\{\mathsf{a}_k\}=\exp\left(N_{\mathrm{G}}\sigma^2\right)-1.$$

Therefore, the standard deviation of the Bloch sphere's radius tends to be $\sqrt{\exp\left(N_{\mathrm{G}}\sigma^2\right)-1}$, which is on the order of $O\left(\sqrt{N_{\mathrm{G}}\sigma^2}\right)$ when $N_{\mathrm{G}}\sigma^2\ll 1$. This agrees with our formal analytical results.

The linear error scaling experienced in the case where no QEM is applied may be interpreted by considering the graphical illustration in Fig. 5.4. Since the computational result $r$ converges exponentially fast to zero as indicated by Proposition 5.1, it deviates from $\widetilde{r}$ linearly when $N_{\mathrm{G}}$ is relatively small, which may be viewed as a lower bound of the computational error. Additionally, the actual evolution of $r$ is also bounded by the

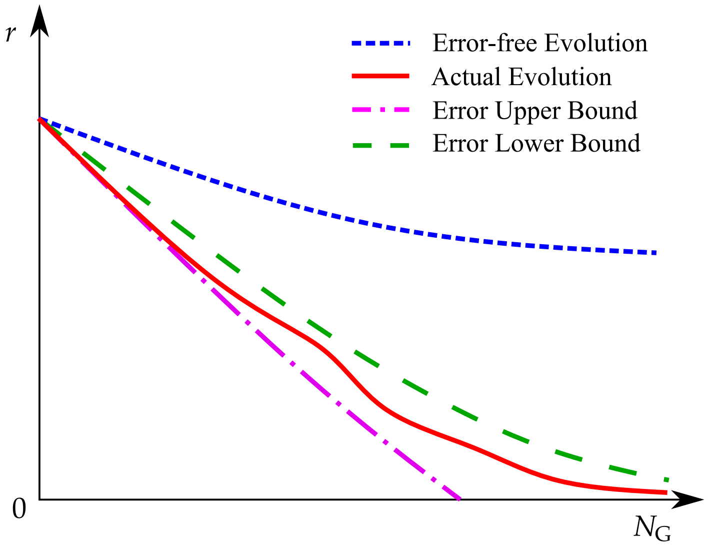


FIGURE 5.4: Demonstration of the evolution of computational result $r$ in the absence of QEM, as a function of $N_G$. The upper and lower bound of computational error correspond to the results in Proposition 5.2 and Proposition 5.1, respectively.

tangent line of it at $N_G = 0$, which gives rise to the error upper-bound in Proposition 5.2.

### 5.4.2 The Accuracy vs. Sampling Overhead Trade-off

If we denote by $\epsilon$ the average error probability of each gate, from the discussion in Section 5.3 we see that the computational error roughly scales as $\Theta(\epsilon)$ when QEM is not applied, whereas it scales as $O(\sqrt{\epsilon})$ when Monte Carlo-based CI-QEM is applied, according to Section 5.3.4. This may be understood by considering the variance of the samples, which is proportional to $\epsilon$. Hence the RMSE is proportional to $\sqrt{\epsilon}$.

Since $\epsilon$ is typically far less than 1, it seems that Monte Carlo-based CI-QEM has a less preferable performance. Nevertheless, it is noteworthy that the error scaling in the CI-QEM-protected case is actually $O(\sqrt{\epsilon N_s^{-1}})$, where the number $N_s$ of effective circuit executions is a configurable parameter. The $O(\sqrt{N_s^{-1}})$ dependency on $N_s$ originates from the fact that the sampling variance scales as $O(N_s^{-1})$. Therefore, our results should not be viewed as indicating the superiority of non-QEM-based solutions. Rather, they should be viewed as a suggestion on the specific selection of $N_s$, in the sense that it should be on the order of $\epsilon$ to ensure the error scaling is as beneficial as that of the family of non-QEM-based solutions.

Similarly, by increasing $N_s$ as a function of $N_G$, one could also improve the error scaling of Monte Carlo-based CI-QEM with the circuit size. Indeed, since the error of Monte Carlo-based QEM scales as $O(\sqrt{N_G N_s^{-1}})$, we can choose an $N_s$ that is proportional to $N_G$ in order to attain a constant error with respect to $N_G$. Note that the exact channel inversion also has a constant error with respect to $N_G$ in the asymptotic limit of

$N_\mathrm{G} \to \infty$. Therefore, using Monte Carlo-based CI-QEM, we could use $N_\mathrm{G}$ times the number of samples to attain the same error scaling as that of QEM based on exact channel inversion. In practical scenarios, however, this may be an excessive sampling overhead. Fortunately, even if we use the same number of samples as that of the exact channel inversion, Monte Carlo-based CI-QEM still exhibits a quadratic error scaling improvement compared to the no-QEM-based case.

### 5.4.3 The Intrinsic Uncertainty of the Computational Results

In the previous discussions, we followed the definition of computational results in (5.5). But even if the gates are decoherence-free, the intrinsic uncertainty of quantum states may impose some randomness on the computational result. To be specific, for a quantum state $\rho$, the variance of a quantum observable $\mathcal{O}$ can be computed as follows [62]:

$$\mathrm{Var}_\rho\{\mathcal{O}\} = \mathrm{Tr}\left\{\mathcal{O}^2\rho\right\} - (\mathrm{Tr}\left\{\mathcal{O}\rho\right\})^2, \tag{5.46}$$

which quantifies the intrinsic uncertainty of the state $\rho$ under the observable $\mathcal{O}$. If the quantum circuit is executed $N_\mathrm{s}$ times, the resultant variance is given by $N_\mathrm{s}^{-1}\mathrm{Var}_\rho\{\mathcal{O}\}$, and hence the mean squared error can be expressed as

$$\mathrm{MSE} = (r - \widetilde{r})^2 + \frac{1}{N_\mathrm{s}} \cdot \mathrm{Var}_\rho\{\mathcal{O}\}. \tag{5.47}$$

We first consider the case where no QEM is applied. Since the observable $\mathcal{M}_\mathrm{ob}$ is typically implemented via a decomposition into Pauli operators in VQAs, its variance may also be decomposed as

$$\mathrm{Var}_{\rho_{N_\mathrm{G}}}\{\mathcal{M}_\mathrm{ob}\} = \sum_{i=1}^{4^n} \frac{1}{2^n}[\boldsymbol{v}_\mathrm{ob}]_i^2 \mathrm{Var}_{\rho_{N_\mathrm{G}}}\{\mathcal{S}_i\}. \tag{5.48}$$

For each Pauli operator, we have

$$\begin{aligned} \mathrm{Var}_{\rho_{N_\mathrm{G}}}\{\mathcal{S}_i\} &= \mathrm{Tr}\left\{\mathcal{S}_i^2\rho_{N_\mathrm{G}}\right\} - (\mathrm{Tr}\left\{\mathcal{S}_i\rho_{N_\mathrm{G}}\right\})^2 \\ &= 1 - (\mathrm{Tr}\left\{\mathcal{S}_i\rho_{N_\mathrm{G}}\right\})^2. \end{aligned} \tag{5.49}$$

Hence we arrive at

$$\begin{aligned} \mathrm{Var}_{\rho_{N_\mathrm{G}}}\{\mathcal{M}_\mathrm{ob}\} &= \sum_{i=1}^{4^n} \frac{1}{2^n}[\boldsymbol{v}_\mathrm{ob}]_i^2 (1 - (\mathrm{Tr}\left\{\mathcal{S}_i\rho_{N_\mathrm{G}}\right\})^2) \\ &= \boldsymbol{v}_\mathrm{ob}^\mathrm{T} \left(\frac{1}{2^n}\boldsymbol{I} - \boldsymbol{V}_{N_\mathrm{G}}^2\right) \boldsymbol{v}_\mathrm{ob}, \end{aligned} \tag{5.50}$$

where we have $\boldsymbol{V}_{N_{\mathrm{G}}} = \mathrm{diag}\left\{\boldsymbol{v}_{N_{\mathrm{G}}}\right\}$. Note that from (5.7) we have $[\boldsymbol{v}_{N_{\mathrm{G}}}]_i^2 \leq 2^{-n}$ for all $i$, hence it follows that

$$0 \leq \mathrm{Var}_{\rho_{N_{\mathrm{G}}}}\{\mathcal{M}_{\mathrm{ob}}\} \leq 1. \tag{5.51}$$

Thus the MSE of the computational result is bounded by

$$(r - \widetilde{r})^2 \leq \mathrm{MSE} \leq (r - \widetilde{r})^2 + \frac{1}{N_{\mathrm{s}}}. \tag{5.52}$$

When quantum circuits are protected by QEM, it has been shown in [37] that if the number of effective executions is $N_{\mathrm{s}}$, the variance equals to that in the case, where no QEM is not applied. Thus the scaling of the total error is on the order of

$$O\left(\sqrt{\frac{\epsilon N_{\mathrm{G}}}{N_{\mathrm{s}}}}\right) + O\left(\sqrt{\frac{1}{N_{\mathrm{s}}}}\right).$$

This implies that, the effect of CI-QEM may not be very significant when $\epsilon N_{\mathrm{G}} \ll 1$. But note that when $\rho_{N_{\mathrm{G}}}$ corresponds to one of the eigenstates of all Pauli operators $i$ having a non-zero coefficient $[\boldsymbol{v}_{\mathrm{ob}}]_i$, we have

$$\mathrm{Var}_{\rho_{N_{\mathrm{G}}}}\{\mathcal{S}_i\} = 1 - \left(\mathrm{Tr}\left\{\mathcal{S}_i \rho_{N_{\mathrm{G}}}\right\}\right)^2 = 0,$$

which follows from the fact that Pauli operators only have eigenvalues of ±1. Therefore, CI-QEM would be more effective when the final state $\rho_{N_{\mathrm{G}}}$ is close to one of these eigenstates.

## 5.5 Numerical Results

In this section, we evaluate the analytical results presented in the previous sections via numerical examples. If not otherwise stated, the following parameters and assumptions will be used throughout the section.

- The number of effective circuit executions is $N_{\mathrm{s}} = 5000$;
- For Monte Carlo-based CI-QEM, we use the same number of samples (i.e., actual circuit executions) as that of CI-QEM based on exact channel inversion;
- The quantum channels modelling the gate imperfections are single-qubit depolarizing channels having gate error probability $10^{-3}$.

### 5.5.1 Rotations Around the Bloch Sphere

We first consider the simplest scenario, where the quantum circuits are constituted of single-qubit gates, because these simple circuits allow us to clearly observe the error

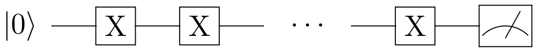


FIGURE 5.5: Circuit implementing repeated Pauli X gates.

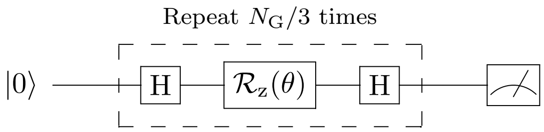


FIGURE 5.6: Circuit implementing repeated $\theta$-rotations around the X-axis.

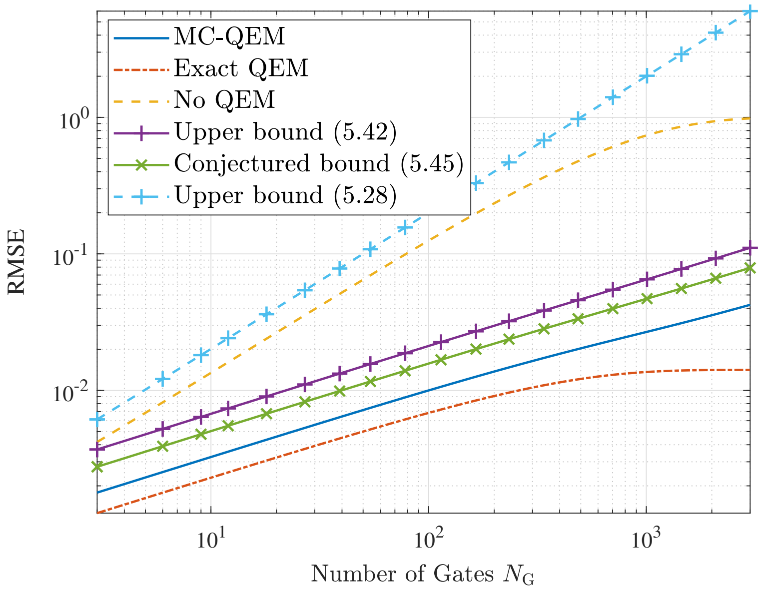


FIGURE 5.7: The RMSE versus $N_G$ of the results computed by quantum circuits consisting of repeated Pauli X gates (as demonstrated in Fig. 5.5).

scaling described in the previous sections. In particular, we consider the circuits shown in Fig. 5.5 and 5.6. The quantum observable $\mathcal{M}_{ob}$ in this example is the Pauli Z operator $\mathcal{Z}$ on the qubit, which satisfies

$$\mathcal{Z}\,|0\rangle = |0\rangle\,,\ \mathcal{Z}\,|1\rangle = -\,|1\rangle\,.$$

The corresponding PTM representation is given by $v_{ob} = [0\;0\;0\;\sqrt{2}]^T$.

For the circuit consisting of repeated Pauli X gates shown in Fig. 5.5, the RMSE of the computational results both with and without QEM protection is demonstrated in Fig. 5.7, as a function of $N_G$. As it can be observed from the figure, when $N_G$ is relatively small, the RMSE of circuits operating without QEM protection grows linearly with $N_G$, while the RMSE of circuits protected by Monte Carlo-based CI-QEM scales as $O(\sqrt{N_G})$. The RMSE of CI-QEM based on exact channel inversion scales as $O(\sqrt{N_G})$ for small $N_G$, but converges to a constant ($\approx \sqrt{N_s^{-1}}$) when $N_G$ is large. Furthermore, when $N_G$ is large, the RMSE of circuits operating without QEM protection converges to a

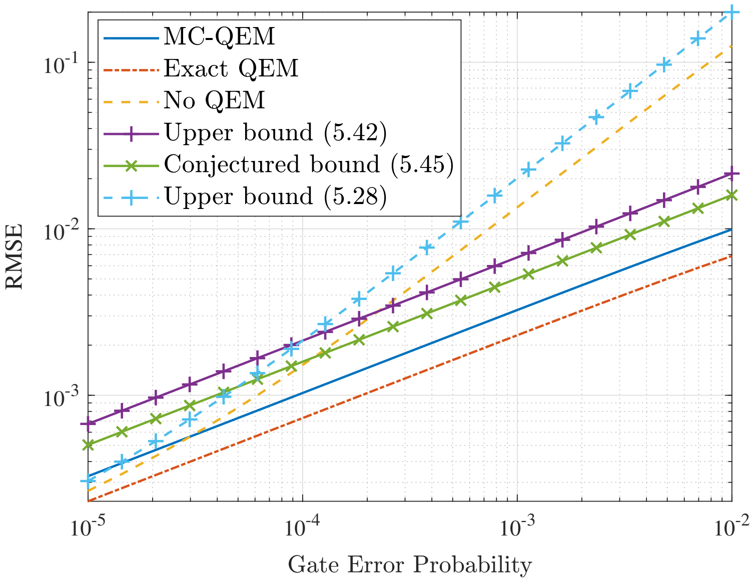


FIGURE 5.8: The RMSE versus gate error probability ($N_{\mathrm{G}} = 10$)of the results computed by quantum circuits consisting of repeated Pauli X gates (as demonstrated in Fig. 5.5).

constant. This agrees with Proposition 5.1, which indicates that their computational results converge to zero regardless of the quantum observable.

The RMSE scalings with respect to the gate error probability $\epsilon$ are shown in Fig. 5.8, where we choose $N_{\mathrm{G}} = 10$, while the number of effective circuit executions, namely $N_{\mathrm{s}} = 5000$, does not vary as the gate error probability increases. It is noteworthy that when $\epsilon$ is small, the RMSE of circuits operating without QEM protection is lower than that of their counterparts protected by QEM. This phenomenon may be understood from our discussion in Section 5.4.2, where we have indicated that the error scaling of CI-QEM-protected circuits is $O(\sqrt{\epsilon N_{\mathrm{s}}^{-1}})$. Compared to the $O(\epsilon)$ scaling of non-QEM-protected circuits, the RMSE may be higher when $\epsilon$ is much smaller than $N_{\mathrm{s}}$. Interestingly, as seen from the figure, the square root scaling with respect to $\epsilon$ becomes preferable to the linear scaling when $\epsilon$ is relatively large.

For the circuit comprising repeated rotations around the $X$-axis, as illustrated in Fig. 5.6, we set $\theta = \pi/256$, and the results are plotted in Fig. 5.9. Observe that the envelope of the RMSE curves exhibit similar scaling behaviours as those in Fig. 5.8, but there are some oscillations. To understand the RMSE oscillations of circuits protected by Monte Carlo-based CI-QEM, from (A.61) we may express the covariance matrix of $\mathbf{v}_k$ as follows

$$\boldsymbol{\Sigma}_k = (\mathbf{1}\mathbf{1}^{\mathrm{T}} + \boldsymbol{\Xi}_k) \otimes \boldsymbol{G}_k \boldsymbol{\Sigma}_{k-1} \boldsymbol{G}_k^{\dagger} + \boldsymbol{\Xi}_k \otimes \boldsymbol{\mu}_k \boldsymbol{\mu}_k^{\mathrm{T}}. \tag{5.53}$$

Note that the term $\boldsymbol{\Xi}_k \otimes \boldsymbol{\mu}_k \boldsymbol{\mu}_k^{\mathrm{T}}$ varies with $k$ under the observable $\mathcal{M}_{\mathrm{ob}} = \mathcal{Z}$, and hence the RMSE is oscillatory.

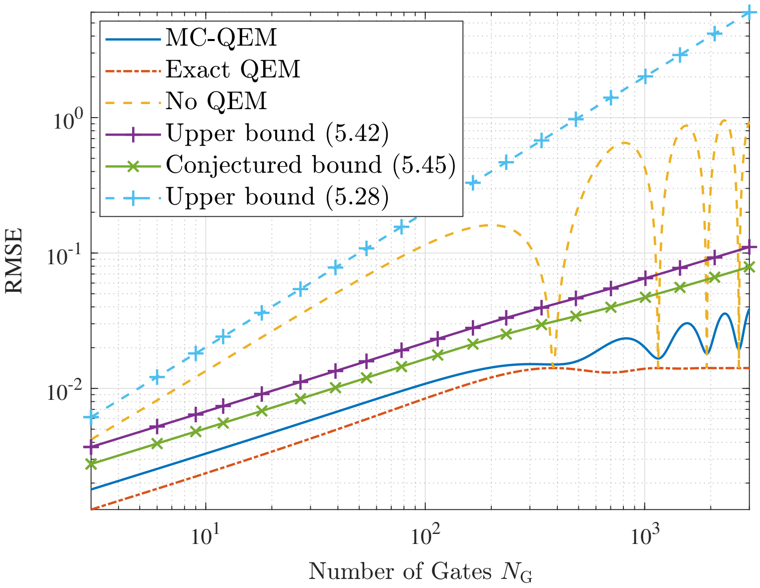


FIGURE 5.9: The RMSE of the results computed by quantum circuits carrying out repeated rotations around the X-axis of the Bloch sphere (as shown in Fig. 5.6), as functions of $N_G$, when $\theta = \pi/256$.

The RMSE oscillation of non-QEM-protected circuits may be better understood by investigating the evolution of the computational result $r$ as $N_G$ increases, which is portrayed in Fig. 5.10. It can be seen that the mean values of the non-QEM-protected circuits fit nicely within the bounds given by Proposition 5.1. Furthermore, the RMSE of CI-QEM-protected circuits is mainly contributed by the variance of the computation results, while the RMSE of circuits not protected by QEM is mainly determined by the mean value, since in the latter case the bias is far larger than the standard deviation. As the dynamic range of the mean values is reduced, by coincidence, there are multiple intersections of the ground truth and the mean values, and thus the computational error of non-QEM-protected circuits oscillates as $N_G$ increases.

Finally, we demonstrate that some non-Pauli channels may also exhibit the $O(\sqrt{N_G})$ error scaling. In particular, we consider amplitude damping channels [107] having the following PTM representation

$$\boldsymbol{C}_{\text{damp}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{1-\gamma} & 0 & 0 \\ 0 & 0 & \sqrt{1-\gamma} & 0 \\ \gamma & 0 & 0 & 1-\gamma \end{pmatrix}, \tag{5.54}$$

where $\gamma$ is the amplitude damping probability. Here, we set the amplitude damping probability to $\gamma = 1 \times 10^{-3}$. The RMSE scalings with respect to the number of gates $N_G$ are shown in Fig. 5.11 for the circuit comprising repeated Pauli X gates, and in Fig. 5.12 for the circuit consisting of repeated $(\pi/256)$ rotations around the $X$-axis. We observe that the curves corresponding to CI-QEM based on exact channel inversion

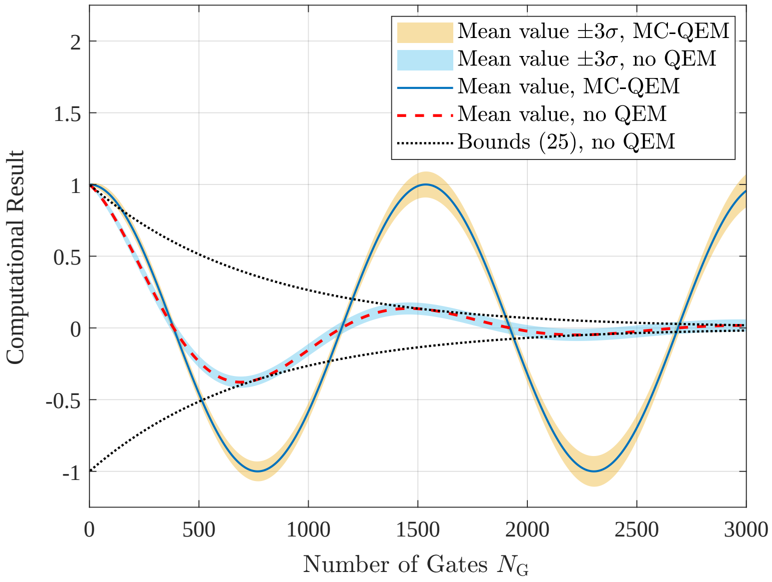


FIGURE 5.10: The computational results of QEM-protected and non-QEM-protected circuits configured for carrying out repeated rotations ($\theta = \pi/256$) around the X-axis of the Bloch sphere (shown in Fig. 5.6), as functions of $N_G$.

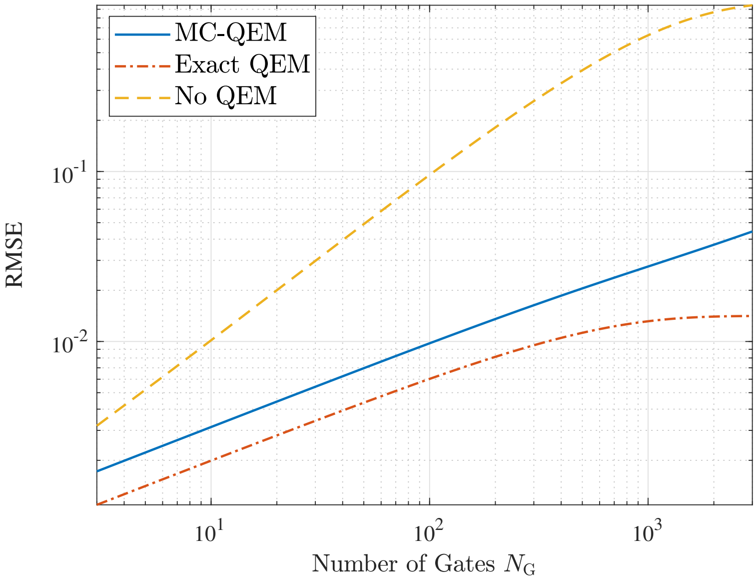


FIGURE 5.11: The RMSE of the results computed by quantum circuits shown in Fig. 5.5, which are contaminated by amplitude damping channels, as functions of $N_G$.

and those corresponding to Monte Carlo-based CI-QEM exhibit the $O(\sqrt{N_G})$ scaling behavior, while the non-QEM-protected curves scale as $O(N_G)$, which is similar to the error scaling under Pauli channels as portrayed in Fig. 5.7 and Fig. 5.9.

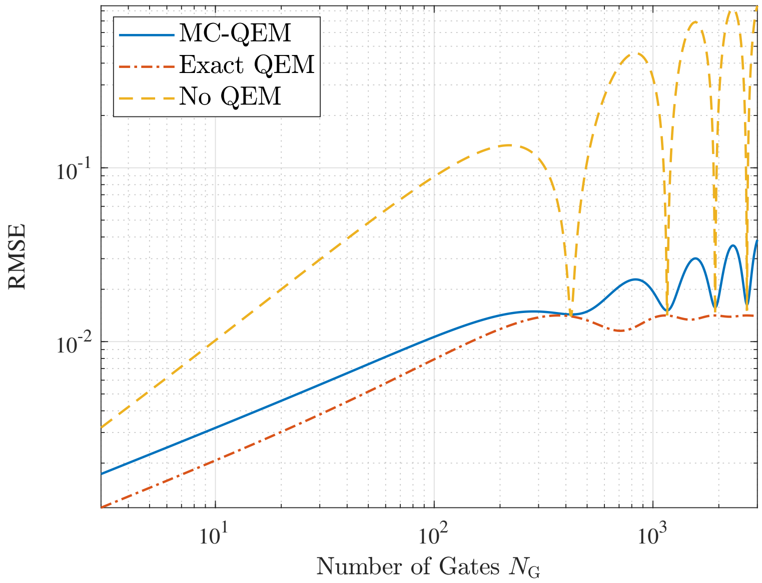


FIGURE 5.12: The RMSE of the results computed by quantum circuits shown in Fig. 5.6, which are contaminated by amplitude damping channels, as functions of $N_\mathrm{G}$.

### 5.5.2 The Quantum Approximate Optimization Algorithm Aided Multi-User Detection

In this subsection, we apply our analytical results to a practical variational quantum algorithm, the quantum approximate optimization algorithm [32], which aims for solving combinatorial optimization problems assuming the following form

$$\max_{z\in\{-1,+1\}^n} \quad \sum_{k=1}^{K} w_k \prod_{i=1}^{n_k} z_{l_{k,i}}, \tag{5.55}$$

where $\boldsymbol{z} = [z_1 \ \ldots \ z_n]^\mathrm{T}$, and $l_{k,i} \in \{1, 2, \ldots, n\}$. In the formulation of the quantum approximate optimization algorithm, the problem (5.55) is transformed into the maximization of $\langle\psi|\,\mathcal{H}\,|\psi\rangle$, where the quantum observable $\mathcal{H}$ is given by

$$\mathcal{H} = \sum_{k=1}^{K} w_k \prod_{i=1}^{n_k} \mathcal{Z}_{l_{k,i}}. \tag{5.56}$$

The trial state $|\psi\rangle$ is prepared using a parametric circuit having an alternating structure, so that

$$|\psi\rangle = e^{-i\beta_P\mathcal{B}} e^{-i\gamma_P\mathcal{H}} \cdots e^{-i\beta_1\mathcal{B}} e^{-i\gamma_1\mathcal{H}} \,|+\rangle^{\otimes n},$$

where $P$ is the number of stages in the alternating circuit, and $\mathcal{B}$ is the "mixing Hamiltonian" [33] given by $\mathcal{B} = \sum_{i=1}^{n} \mathcal{X}_i$. The parameters $\boldsymbol{\beta} = [\beta_1 \ \ldots \ \beta_P]^\mathrm{T}$ and $\boldsymbol{\gamma} = [\gamma_1 \ \ldots \ \gamma_P]^\mathrm{T}$ are typically obtained using via an optimization procedure implemented on classical computers [34]. For the purpose of this treatise, here we do not optimize the parameters, but use the following (suboptimal) adiabatic configuration [185] instead

$$\gamma_k = kP^{-1},\ \ \beta_k = 1 - kP^{-1}.$$

We consider the multiuser detection problem of wireless communications [186]. In particular, assuming that the modulation scheme is BPSK, in a spatial division multiple access system, the signal received at a base station equipped with $m$ antennas from $n$ single-antenna uplink transmitters may be expressed as

$$\mathbf{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{\omega},$$

where $\boldsymbol{H} \in \mathbb{R}^{m\times n}$ denotes the channel, $\boldsymbol{x} \in \{-1,+1\}^n$ represents the transmitted signal, and $\boldsymbol{\omega} \in \mathbb{R}^m$ is the noise. We assume here that the noise is i.i.d. Gaussian. Hence the maximum likelihood estimate of $\boldsymbol{x}$ is given by

$$\hat{\boldsymbol{x}}_{\mathrm{ML}} = \mathop{\arg\max}_{\boldsymbol{x}\in\{-1,+1\}^n}\ 2(\boldsymbol{H}\mathbf{y})^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{x}.$$

This may be further reformulated as the maximization of the quadratic form $\langle\psi|\,\mathcal{H}\,|\psi\rangle$, where

$$\mathcal{H} = \frac{1}{Z}\left(\sum_{i=1}^{n}[\boldsymbol{H}^{\mathrm{T}}\mathbf{y}]_i\mathcal{Z}_i - \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}[\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}]_{i,j}\mathcal{Z}_i\mathcal{Z}_j\right), \tag{5.57}$$

and $Z$ is a normalizing coefficient ensuring that the quantum observable $\mathcal{H}$ satisfies our Assumption 3.

In this illustrative example, we consider the case where $m = n = 4$, and $[\boldsymbol{\omega}]_i \sim \mathcal{N}(0, 0.0631), \forall i$, such that the signal-to-noise ratio is 12dB. We assume furthermore that the channels between each pair of antennas are uncorrelated non-dispersive Rayleigh channels, hence the entries of the channel $\boldsymbol{H}$ are i.i.d. Gaussian variables with zero mean and variance $m^{-1}$ [187]. For the quantum circuits, we choose gate error probability $\epsilon = 3\times 10^{-4}$. Under these assumptions, the RMSE scalings with respect to $P$ of non-QEM-protected circuits and that of circuits protected by Monte Carlo-based CI-QEM are portrayed in Fig. 5.13. It can be observed that the non-QEM-protected circuits exhibit an $O(P)$ scaling, while the CI-QEM-protected circuits exhibit an $O(\sqrt{P})$ scaling, as indicated by Propositions 5.2 and 5.4, respectively.

To illustrate the evolution of the computational results during the execution of circuits, we plot the objective function values (i.e., $\langle\psi|\,\mathcal{H}\,|\psi\rangle$) computed at each stage $k$ of the circuits in Fig. 5.14, for the case where $P = 225$. Note that the results computed by the QEM-protected circuits converge monotonically towards the optimum, for which the main source of error is the variance. By contrast, for the non-QEM-protected circuits,

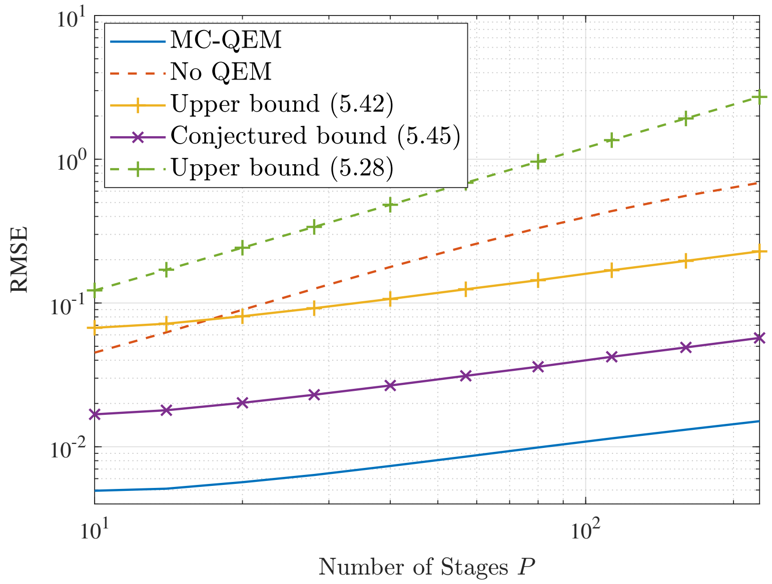


FIGURE 5.13: The RMSE of the results computed by QEM-protected and non-QEM-protected circuits implementing the quantum approximate optimization algorithm based on (5.57), as functions of the number of stages $P$.

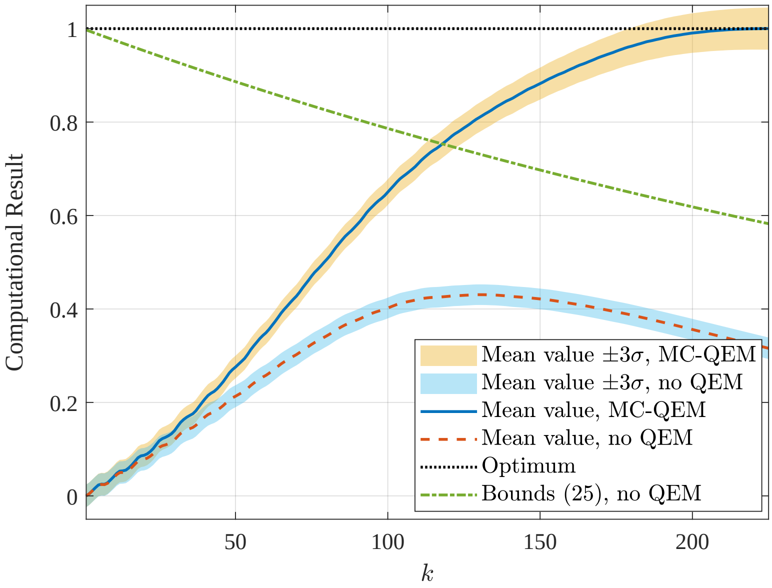


FIGURE 5.14: The objective function values computed at the $k$-th stage of the quantum approximate optimization algorithm (for which $P = 225$) implemented based on (5.57).

the results were on the right track for $k < 100$, but soon they deviate from their QEM-protected counterparts, and start to converge to zero. In this example, the bound (5.25) is not as tight as it was in Section 5.5.1, but it still indicates that the dynamic range of the results computed by non-QEM-protected circuits decays exponentially as $k$ increases.

## 5.6 Conclusions and Future Research

### 5.6.1 Conclusions

The trade-off between the computational overhead and the error scaling behaviour of both quantum circuits protected by Monte Carlo-based CI-QEM and their non-QEM-protected counterparts was investigated. As for the non-QEM-protected circuits, we have shown that the dynamic range of the noisy computational results shrinks exponentially as the number of gates $N_{\mathrm{G}}$ increases, implying a linear error scaling with $N_{\mathrm{G}}$. By contrast, the error scales as the square root of $N_{\mathrm{G}}$ in the presence of Monte Carlo-based QEM, at the same computational cost as that of CI-QEM based on exact channel inversion. Moreover, the error scaling of Monte Carlo-based QEM can be further improved with increased computational cost. We have also demonstrated the analytical results both for low-complexity examples and for a more practical example of the quantum approximate optimization algorithm employed for multi-user detection in wireless communications. It may be an interesting future research direction to apply the results to other practical examples, or verify them using experimental approaches.

### 5.6.2 Future Research

- In this chapter, we have assumed that the channel inversion is implemented by sampling directly from the quasi-probability representation of the inverse channels. In fact, other sampling strategies might also be useful when we consider the Monte Carlo-based implementation of inverse channels. For example, let us consider the inversion of a depolarizing channel, characterized by its Pauli transfer matrix given by

  $$\boldsymbol{C} = \mathrm{diag}\{1, 0.99, 0.99, 0.99\}. \tag{5.58}$$

  If we use the standard Monte Carlo sampling, we would sample from the distribution of $[0.9925,\ 0.0025,\ 0.0025,\ 0.0025]^{\mathrm{T}}$, which is obtained by renormalizing the quasi-probability representation. However, when the number of samples is not sufficiently large, for example, if we use $N_{\mathrm{s}} = 1000$ samples, the events corresponding to small probabilities such as 0.0025 would rarely occur, leading to a large variance. Alternatively, one may deliberately allocate more samples to these rare events to reduce the variance. The correct expectation values may then be obtained upon multiplying these samples by a weighting factor (which is less than 1) to these samples. However, this sample re-allocation might also increase the variance corresponding to the error-free event. Therefore, an interesting future research topic would be finding the optimal sampling allocation strategy in terms of the overall sample variance.

- The stabilizer measurements of coded gates may provide valuable information about the channel condition. In light of this, circuit execution results may be used simultaneously for both channel estimation and CI-QEM, which may further reduce the overall sampling overhead. Moreover, this may also improve the capability of channel estimation results to accommodate the channel variations vs. time.

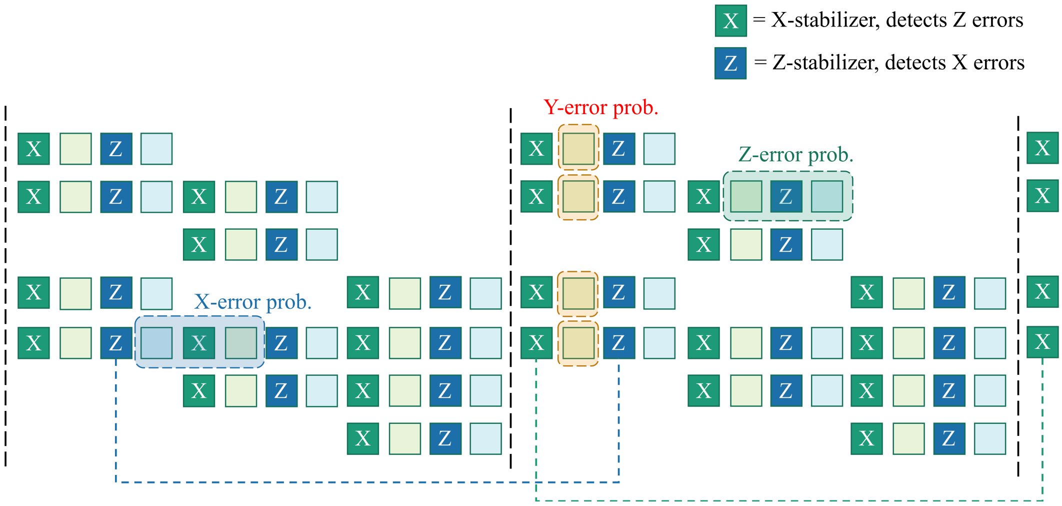


FIGURE 5.15: The "differential detection" method in code-aided CI-QEM.

Specifically, the quantum channels concerning the quality of the stabilizer check circuits may be estimated relying on the measured syndromes, using a technique similar to the concept of "differential detection" in wireless communications, as portrayed in Fig. 5.15. Explicitly, the stabilizer-induced channels are represented by the boxes following the corresponding stabilizer checks. As may be seen from the figure, we may repeat the stabilizer checks for several rounds. This helps us to extract the information about the quantum channel corresponding to the stabilizer checks by subtracting the syndrome measured in different rounds of stabilizer checks, as represented by the broken lines in Fig. 5.15. After solving a linear system of equations based on these subtracted measurements, we may then obtain rough estimates of the error probability of stabilizer checks. Finding an efficient way of constructing these equations and refining the rough estimates may be an important future research direction.

# Chapter 6

# Permutation Symmetry Based QEM

We have introduced symmetry-based QEM in Section 3.3.2. Specifically, the first symmetry-based QEM method, namely symmetry verification, is only applicable when the type of state symmetry characterized by the stabilizer group is precisely known in advance. By contrast, the recently proposed virtual distillation (VD) concept [65, 66] relying on the permutation symmetry of quantum states has the advantage that the symmetry of the states can be easily manipulated. To elaborate, VD prepares multiple copies of the same quantum state, and filters out the components in the states that are not identical across all copies, as portrayed in Fig. 6.1. The observables are then measured on one of the copies. The error mitigation capability can be flexibly strengthened by increasing the number of copies, but naturally, at the cost of higher qubit overhead.

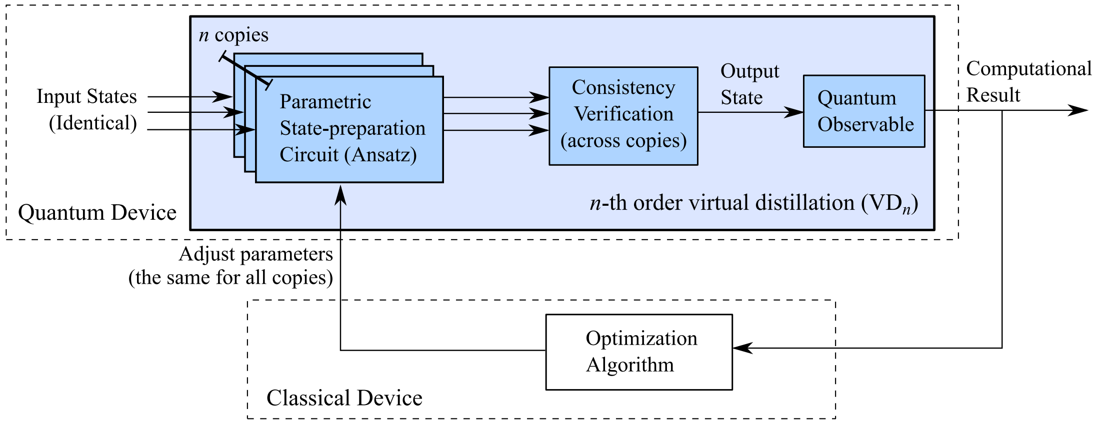


FIGURE 6.1: An $n$-th order virtual distillation method (relying on $n$ copies of the parametric state-preparation circuits) applied to a variational quantum algorithm.

From the spectral analysis perspective of quantum states, when the noise is not extremely strong, the dominant eigenvector of the output state serves as a good approximation of the ideal noise-free output state [65]. In this sense, VD may be implemented as a high-pass filter in the spectral domain, since its output would tend to the dominant

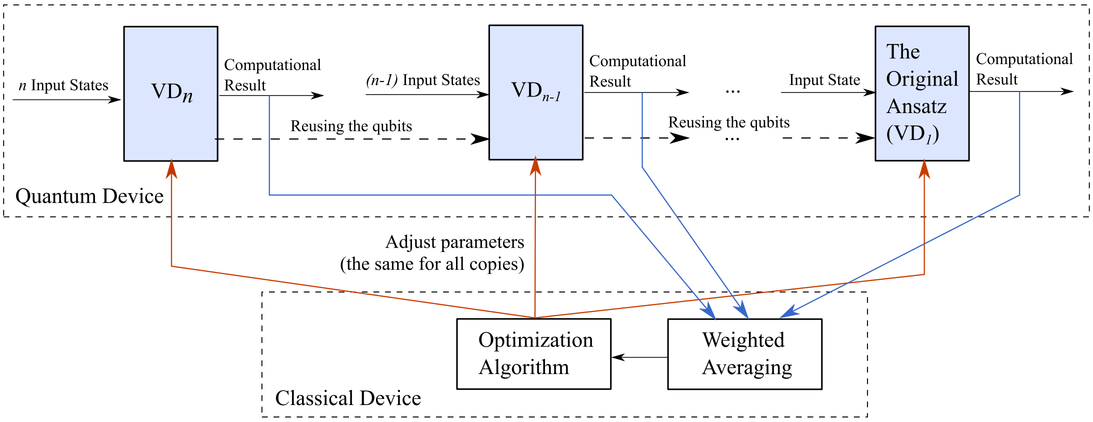


FIGURE 6.2: An $n$-th order permutation filter proposed in this chapter applied to a variational quantum algorithm.

eigenvector as the number of copies $N$ tends to infinity, according to our discussion in Section 3.3.2.2.

In this chapter, we first discuss the rationale of the VD method in more detail, and then generalize this idea by proposing a general framework for designing optimal filters in the spectral domain of quantum states. These filters assume a similar form as the finite impulse response (FIR) filters widely used in classical signal processing tasks, by computing a weighted average over the outputs of multiple virtual distillation circuits of different orders, as shown in Fig. 6.2.

The rest of this chapter is organized as follows. In Section 6.1 we provide a detailed introduction to permutation symmetry and virtual distillation. In Section 6.2, we describe the permutation filter as well as its design algorithm. Then, in Section 6.3 we analyze the error-reduction performance of permutation filters. The results are further illustrated using numerical results in Section 6.4. Finally, we conclude the paper in Section 6.5.

## 6.1 Permutation Symmetry and Virtual Distillation

The permutation-based quantum error mitigation philosophy is inspired by the concept of permutation tests, which constitute generalizations of the swap test [188]. As portrayed in Fig. 6.3, the swap test is implemented by controlled-SWAP gates. It is widely employed for evaluating the overlap between a pair of quantum states $\rho$ and $\sigma$, since the expected value of the measurement outcome is given by $\mathrm{Tr}\{\rho\sigma\}$. Naturally, when we have two copies of the same state $\rho$, we may compute $\mathrm{Tr}\{\rho^2\}$ using the swap test.

The permutation tests, exemplified by the cyclic-shift test [189], may be implemented using quantum circuits taking the form shown in Fig. 6.4. As a generalization of the

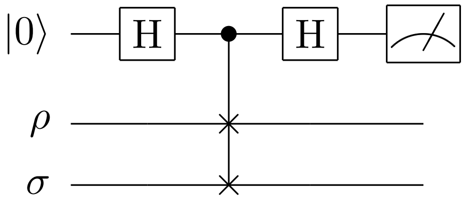


FIGURE 6.3: Schematic of the swap test.

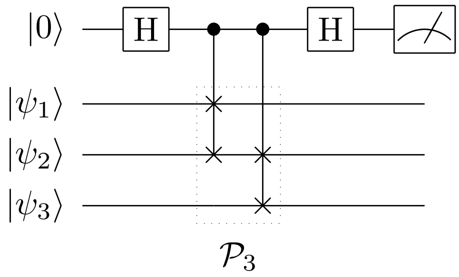


FIGURE 6.4: Schematic of the permutation test.

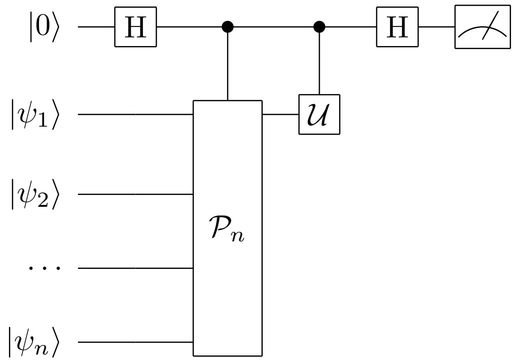


FIGURE 6.5: Schematic of Koczor's method.

swap gate, an $n$-th order cyclic-shift circuit $\mathcal{P}_n$ taking an input of $n$ pure states $|\psi_1, \psi_2, \ldots, \psi_n\rangle$ would output a shifted state $|\psi_2, \psi_3, \ldots, \psi_n, \psi_1\rangle$. Note that the swap gate may be viewed as a specific case of cyclic-shift circuit, since it is equivalent to $\mathcal{P}_2$. Similar to the swap test, one may show that the expectation value of the outcome in an $n$-th order cyclic-shift test is given by $\mathrm{Tr}\{\rho^n\}$ [65], when the inputs are represented by $n$ copies of the same mixed state $\rho$.

Typically, when quantum circuits are contaminated by decoherence, the output state would approximately take the following form

$$\rho = \lambda_1 |\psi\rangle \langle\psi| + \sum_{i=2}^{2^{N_q}} \lambda_i |\psi_i\rangle \langle\psi_i| , \tag{6.1}$$

where $|\psi_i\rangle$ denotes the eigenvector associated with the $i$-th largest eigenvalue of $\rho$, and $|\psi\rangle = |\psi_1\rangle$ is the dominant eigenvector, which approximates the noise-free output

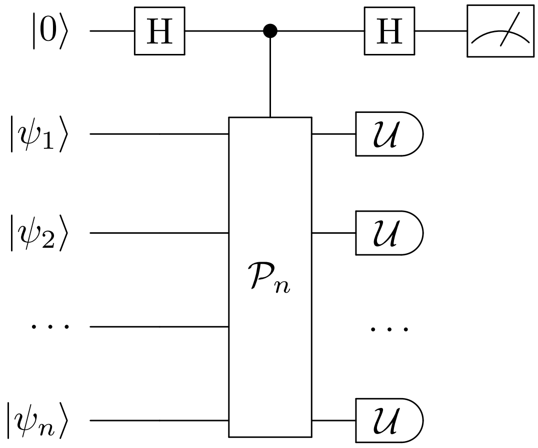


FIGURE 6.6: The VD in [66]

state [65, 123]. Inspired by these observations, Koczor [65] proposed the concept of permutation-based quantum error mitigation (which has later been generalized to the concept of VD [66]), as portrayed in Fig. 6.5. Compared to the permutation test shown in Fig. 6.4, it may be observed that the output of the VD circuit for a given unitary observable $\mathcal{U}$ is given by

$$\tilde{y}_{\mathrm{VD}}^{(n)} = \mathrm{Tr}\left\{\rho^n \mathcal{U}\right\}, \tag{6.2}$$

where $n$ is the order of the circuit $\mathcal{P}_n$, and we will also refer to it as the order of VD. Another implementation yielding the same result as in (6.2) is proposed in [66], as shown in Fig. 6.6. This implementation enables simultaneous measurement of multiple compatible observables, and thus reduces the total number of circuit repetitions. Note that all Pauli strings are unitary observables, hence they can be nicely fit into this framework. Next, upon replacing the observable $\mathcal{U}$ by the identity operator $\mathcal{I}$ (i.e., the original $n$-th order permutation test), one may also compute $\mathrm{Tr}\left\{\rho^n\right\}$, and obtain the final result

$$y_{\mathrm{VD}}^{(n)} = \frac{\tilde{y}_{\mathrm{VD}}^{(n)}}{\mathrm{Tr}\left\{\rho^n\right\}} = \frac{\mathrm{Tr}\left\{\rho^n \mathcal{U}\right\}}{\mathrm{Tr}\left\{\rho^n\right\}}. \tag{6.3}$$

The accuracy of this normalization procedure may be further improved by replacing $\mathrm{Tr}\left\{\rho^n\right\}$ with $\lambda_1^n$. However, $\lambda_1$ is typically not known prior to the computation, and is also difficult to be computed exactly from the observations. By contrast, $\mathrm{Tr}\left\{\rho^n\right\}$ is readily obtainable by observing the identity operator.

Note that

$$\tilde{y}_{\mathrm{VD}}^{(n)} = \lambda_1^n \langle\psi|\,\mathcal{U}\,|\psi\rangle + (1-\lambda_1)^n \sum_{i=2}^{2^{N_{\mathrm{q}}}} p_i^n \langle\psi_i|\,\mathcal{U}\,|\psi_i\rangle\,, \tag{6.4}$$

where $p_i = \lambda_i(1-\lambda_1)^{-1}$ satisfies $\sum_{i=2}^{N_{\mathrm{q}}} p_i = 1$. When $\lambda_1$ is far larger than the other eigenvalues, it becomes clear from (6.4) that the term $(1-\lambda_1)^n$ decreases much more rapidly with $n$ than $\lambda_1^n$. Hence the contribution of the undesired components $|\psi_i\rangle$, $i > 1$ to the final computation result is substantially reduced by VD.

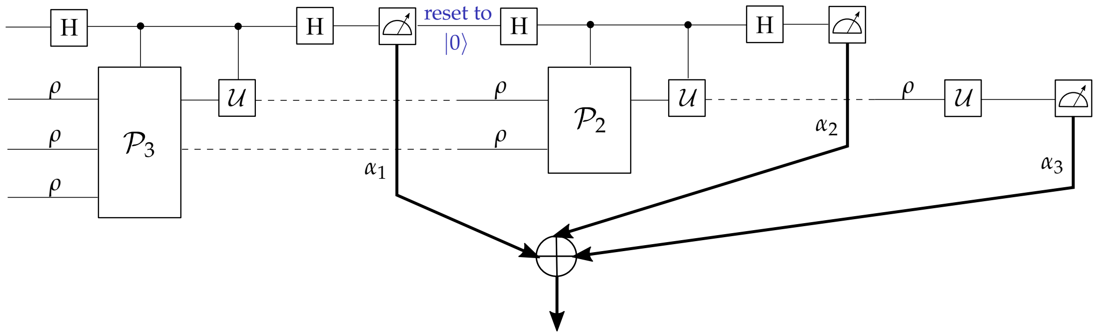


FIGURE 6.7: Schematic of a third-order permutation filter $\mathrm{Tr}\{(\alpha_1\rho^3 + \alpha_2\rho^2 + \alpha_3\rho)\mathcal{U}\}$.

## 6.2 Permutation Filters

In this section, we propose a generalized version of virtual distillation, which will be referred to as "permutation filters". A third-order permutation filter is portrayed in Fig. 6.7. As it may be observed from the figure, the third-order filter consists of the third-order and the second-order VD circuits. In general, an $n$-th order permutation filter would contain all the $m$-th order VD circuits, where $m = 2, 3, \ldots, n$. Note that these circuits can be activated one after the other by reusing the same qubit resources, since the post-processing stage only involves a weighted averaging of the measured outcomes, which are classical quantities.

Formally, an $N$-th order permutation filter may be expressed as an $N$-th order polynomial of the input state $\rho$ formulated as

$$\mathcal{F}_{\boldsymbol{\alpha}}(\rho) = \sum_{n=1}^{N} \alpha_{N-n+1}\rho^n, \tag{6.5}$$

where $\boldsymbol{\alpha} = [\alpha_1\ \alpha_2\ \ldots\ \alpha_N]^{\mathrm{T}} \in \mathbb{R}^N$. Correspondingly, the eigenvalues of the output state are thus given by

$$h_{\boldsymbol{\alpha}}(\lambda) = \sum_{n=1}^{N} \alpha_{N-n+1}\lambda^n. \tag{6.6}$$

Observe that the function $h_{\boldsymbol{\alpha}}(\lambda)$ may be viewed as the "spectral response" of the filter, resembling the frequency response of conventional filters used in classical signal processing tasks. The final computational result with respect to an observable $\mathcal{U}$ is given by

$$y_{\text{filter}}^{(N)}(\mathcal{U}) = \frac{\mathrm{Tr}\left\{\mathcal{F}_{\boldsymbol{\alpha}}(\rho)\mathcal{U}\right\}}{\mathrm{Tr}\left\{\mathcal{F}_{\boldsymbol{\alpha}}(\rho)\right\}}. \tag{6.7}$$

The reason that we do not include the constant term $\alpha_{N+1}$ in (6.5) is that it does not contribute to the final computational results in (6.7) for most practical applications. To elaborate, consider the Pauli string decomposition of observables used in variational quantum algorithms, taking the following form

$$\mathcal{H} = \sum_{i=1}^{4^{N_\mathrm{q}}} w_i \mathcal{S}_i^{(N_\mathrm{q})}, \tag{6.8}$$

where $\mathcal{S}_i^{(N_\mathrm{q})}$ denotes the $i$-th Pauli string acting upon $N_\mathrm{q}$ qubits, given by

$$\mathcal{S}_i^{(N_\mathrm{q})} = \bigotimes_{j=1}^{N_\mathrm{q}} \mathcal{S}^{(1)}_{\mathrm{digit}(i,j)+1}, \tag{6.9}$$

where $\mathrm{digit}(i, j)$ represents the $j$-th digit of $i$ when treated as a base-4 number. Since the single-qubit Pauli operators except for the identity have a trace of zero, we have $\mathrm{Tr}\left\{\mathcal{U}\right\} = 0$ for every Pauli string $\mathcal{U}$. Therefore, even if we include the constant coefficient $\alpha_{N+1}$ in our filter, it will not contribute to the final result, since we have:

$$\alpha_{N+1}\mathrm{Tr}\left\{\rho^0\mathcal{U}\right\} = 0. \tag{6.10}$$

As for the term involving the identity operator, we could simply account for it by adding a constant to the final computational result, since $\mathrm{Tr}\left\{\rho\right\} = 1$ always holds.

It is often convenient to design filters under an alternative parametrization, namely the pole-zero representation widely used in classical signal processing theory. To elaborate, in classical signal processing theory, filters are represented by a ratio between two polynomials in the complex frequency domain. "Poles" refers to the roots of the denominator polynomial, while "zeros" refer to the roots of the numerator polynomial. When considering "FIR-like" filters taking the form (6.5) (since there is no denominator in this formula), there are only zeros but no poles. Observe from (6.5) that the first zero is at $\beta = 0$ due to the lack of the constant term. Upon denoting the remaining zeros by $\boldsymbol{\beta} = [\beta_1 \ \ldots \ \beta_{N-1}]^\mathrm{T}$, we have

$$\mathcal{F}_{\boldsymbol{\beta}}(\rho) = \rho \prod_{n=1}^{N-1} (\rho - \beta_n \boldsymbol{I}), \tag{6.11}$$

and

$$h_{\boldsymbol{\beta}}(\lambda) = \lambda \prod_{n=1}^{N-1} (\lambda - \beta_n). \tag{6.12}$$

The relationship between $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is

$$\boldsymbol{\alpha} = \mathop{\bigstar}_{n=1}^{N-1} [1, \ -\beta_n]^\mathrm{T}, \tag{6.13}$$

where we define $\bigstar_{n=1}^{K} v_n := v_1 \star v_2 \star \ldots \star v_K$, and $\star$ denotes the discrete convolution given by

$$[\boldsymbol{x} \star \boldsymbol{y}]_n = \sum_{i=\max\{1,k+1-n\}}^{\min\{k,m\}} x_i y_{k-i+1},$$

where $\boldsymbol{x} \in \mathbb{R}^m$, $\boldsymbol{y} \in \mathbb{R}^n$, and $\boldsymbol{x} \star \boldsymbol{y} \in \mathbb{R}^{m+n-1}$. Without loss of generality, we assume that

$$\beta_1 \leq \beta_2 \leq \ldots \leq \beta_{N-1}. \tag{6.14}$$

### 6.2.1 The Performance Metric of Permutation Filter Design

For a given observable $\mathcal{U}$, we would hope to minimize the estimation error

$$\begin{aligned} \epsilon_{\mathcal{U}}(\boldsymbol{\beta}) &= \left| y_{\text{filter}}^{(N)}(\mathcal{U}) - \langle \psi | \mathcal{U} | \psi \rangle \right| \\ &= \left| \frac{\sum_{i=1}^{2^{N_\text{q}}} h_{\boldsymbol{\beta}}(\lambda_i) \langle \psi_i | \mathcal{U} | \psi_i \rangle}{\sum_{i=1}^{2^{N_\text{q}}} h_{\boldsymbol{\beta}}(\lambda_i)} - \langle \psi | \mathcal{U} | \psi \rangle \right| \\ &= \left| \frac{\frac{1}{h_{\boldsymbol{\beta}}(\lambda_1)} \sum_{i=2}^{2^{N_\text{q}}} h_{\boldsymbol{\beta}}(\lambda_i) \left( \langle \psi_i | \mathcal{U} | \psi_i \rangle - \langle \psi | \mathcal{U} | \psi \rangle \right)}{1 + [h_{\boldsymbol{\beta}}(\lambda_1)]^{-1} \sum_{i=2}^{2^{N_\text{q}}} h_{\boldsymbol{\beta}}(\lambda_i)} \right|. \end{aligned} \tag{6.15}$$

However, in a typical variational quantum algorithm, a large number of unitary observables $\mathcal{U}_1, \ldots, \mathcal{U}_{N_\text{ob}}$ would have to be evaluated. In light of this, we consider the minimization of the following upper bound

$$\begin{aligned} \epsilon_{\mathcal{U}}(\boldsymbol{\beta}) &\leq \epsilon(\boldsymbol{\beta}) \\ &= \frac{2}{h_{\boldsymbol{\beta}}(\lambda_1)} \left\| \boldsymbol{h}_{\boldsymbol{\beta}}(\tilde{\boldsymbol{\lambda}}) \right\|_1, \end{aligned} \tag{6.16}$$

where $\tilde{\boldsymbol{\lambda}} = [\boldsymbol{\lambda}]_{2:2^{N_\text{q}}}$, and $\boldsymbol{\lambda} = [\lambda_1 \ \ldots \ \lambda_{2^{N_\text{q}}}]^\text{T}$.

If we know *a priori* the distribution of $\tilde{\boldsymbol{\lambda}}$, or in other words, the spectral density of $\rho$ (excluding the dominant eigenvalue), we may directly minimize the cost function $\epsilon(\boldsymbol{\beta})$ as follows:

$$\begin{aligned} &\min_{\boldsymbol{\beta}} \ \epsilon(\boldsymbol{\beta}), \\ &\text{s.t.} \ \ \boldsymbol{\beta} \in \mathscr{B}, \ (6.14), \end{aligned} \tag{6.17}$$

where $\epsilon(\boldsymbol{\beta})$ can be rewritten as

$$\epsilon(\boldsymbol{\beta}) = \frac{1}{\lambda_1 \prod_{n=1}^{N-1} (\lambda_1 - \beta_n)} \int_{\lambda_\text{m}}^{1} \left| \lambda \prod_{n=1}^{N-1} (\lambda - \beta_n) \right| f(\lambda) \mathrm{d}\lambda,$$

$\lambda_{\mathrm{m}} > 0$ denotes the minimum value of $\lambda$, and $f(\lambda)$ denotes the spectral density. The feasible region $\mathscr{B}$ is given by

$$\mathscr{B} = \{\boldsymbol{\beta} | \boldsymbol{\beta} \succcurlyeq \mathbf{0}, \beta_1 \leq \beta_2 \leq \ldots \leq \beta_{N-1}\}.$$

For most practical scenarios, we have $\beta_i \ll \lambda_1$, hence $\epsilon(\boldsymbol{\beta})$ may be approximated as

$$\epsilon(\boldsymbol{\beta}) \approx \tilde{\epsilon}(\boldsymbol{\beta}) = \int_{\lambda_{\mathrm{m}}}^{1} \left| \lambda \prod_{n=1}^{N-1} (\lambda - \beta_n) \right| f(\lambda) \mathrm{d}\lambda, \tag{6.18}$$

since the optimal solution is hardly affected by the denominator. Note that $h_{\boldsymbol{\beta}}(1)$ is always positive, hence we may further simplify the approximated objective function as follows:

$$\begin{aligned} \tilde{\epsilon}(\boldsymbol{\beta}) &= \int_{\lambda_{\mathrm{m}}}^{1} \left| G_{\boldsymbol{\beta}}(\lambda) \right| \mathrm{d}\lambda \\ &= \sum_{i=0}^{N-1} (-1)^i \int_{\beta_{N-i-1}}^{\beta_{N-i}} G_{\boldsymbol{\beta}}(\lambda) \mathrm{d}\lambda, \end{aligned} \tag{6.19}$$

where $G_{\boldsymbol{\beta}}(\lambda) = f(\lambda)\lambda \prod_{n=1}^{N-1}(\lambda - \beta_n)$, and additionally we define $\beta_N = 1$ and $\beta_0 = \lambda_{\mathrm{m}}$.

### 6.2.2 Practical Permutation Filter Design Algorithms

When $f(\lambda)$ is known exactly, we may directly solve the optimization problem discussed in the previous subsection. However, for practical applications, $f(\lambda)$ is never known precisely; it has to be estimated from observations. In this treatise, we fit Pareto distribution [190, 191] to $f(\lambda)$ which is formulated as:

$$f(\lambda) = k\lambda_{\mathrm{m}}^k \lambda^{-(k+1)}, \tag{6.20}$$

where $k > 2$ is a shape parameter.

The reason for using the Pareto distribution is two-fold. First of all, it approximates our empirical observations concerning the output spectra of noisy quantum circuits quite closely. Secondly, it fits nicely with the polynomial form of the permutation filter, making the design problem more tractable. Specifically, under the parametrization of the Pareto distribution, the indefinite integral of $G_{\boldsymbol{\beta}}(\lambda)$ can be explicitly calculated as follows:

$$\begin{aligned} \tilde{G}_{\boldsymbol{\alpha}}(\lambda) &= \frac{1}{k\lambda_{\mathrm{m}}^k} \int G_{\boldsymbol{\beta}}(\lambda) \mathrm{d}\lambda \\ &= \int \lambda^{-k} \prod_{n=1}^{N-1} (\lambda - \beta_n) \mathrm{d}\lambda \\ &= \sum_{n=1}^{N} \frac{\alpha_{N-n+1}}{n-k} \cdot \lambda^{n-k}. \end{aligned} \tag{6.21}$$

The definite integrals in (6.19) can then be obtained as

$$\int_{\beta_i}^{\beta_{i+1}} G_{\boldsymbol{\beta}}(\lambda)\mathrm{d}\lambda = k\lambda_{\mathrm{m}}^{k}\left(\tilde{G}_{\boldsymbol{\alpha}}(\beta_{i+1}) - \tilde{G}_{\boldsymbol{\alpha}}(\beta_i)\right). \tag{6.22}$$

Note that for an $N$-th order permutation filter, we may obtain $N-1$ observations $\boldsymbol{m} = [m_1 \ \dots \ m_{N-1}]^{\mathrm{T}}$ where $m_i = \operatorname{Tr}\left\{\rho^{i+1}\right\}$. These observations can be used to fit the Pareto distribution to $f(\lambda)$ using the method of moments [92]. For example, when $N = 3$, the equations of moments are given by

$$\begin{aligned} \frac{1-\hat{\lambda}_1(\boldsymbol{m})}{2^{N_{\mathrm{q}}}-1} &= \frac{k\lambda_{\mathrm{m}}}{k-1}, \\ \frac{m_1-\hat{\lambda}_1(\boldsymbol{m})^2}{2^{N_{\mathrm{q}}}-1} &= \frac{k\lambda_{\mathrm{m}}^2}{k-2}, \end{aligned} \tag{6.23}$$

where $\hat{\lambda}_1(\boldsymbol{m})$ is an estimate of $\lambda_1$. Here, the quantities $\frac{1-\hat{\lambda}_1(\boldsymbol{m})}{2^{N_{\mathrm{q}}}-1}$ and $\frac{m_1-\hat{\lambda}_1(\boldsymbol{m})^2}{2^{N_{\mathrm{q}}}-1}$ are estimates of the mean value and the variance of the spectrum, respectively. We do not use the conventional sample mean and variance, because the eigenvalues cannot be sampled directly. A natural choice of $\hat{\lambda}_1(\boldsymbol{m})$ for an $N$-th order filter is

$$\hat{\lambda}_1(\boldsymbol{m}) = \|\boldsymbol{\lambda}\|_N = (m_{N-1})^{\frac{1}{N}}, \tag{6.24}$$

which is asymptotically exact as $N \to \infty$, since $\lambda_1 = \|\boldsymbol{\lambda}\|_\infty$.

Using the equations of moments in (6.23), we may then estimate the unknown parameters $k$ and $\lambda_{\mathrm{m}}$. However, for the $N = 2$ case, the method of moments would encounter an identifiability problem, since the number of observations (one) is less than the number of parameters (two). Fortunately, we may obtain the closed-form solution of $\beta_1$ as follows:

$$\beta_1 = \lambda_{\mathrm{m}}\left[2(1+\lambda_{\mathrm{m}}^{k-1})^{-1}\right]^{\frac{1}{k-1}}, \tag{6.25}$$

which is obtained by taking the derivative of $\tilde{\epsilon}(\boldsymbol{\beta})$ with respect to $\beta_1$ and setting it to zero. For $k \geq 2$, $\beta_1$ can be closely approximated by

$$\beta_1 \approx \mu = k\lambda_{\mathrm{m}}(k-1)^{-1}, \tag{6.26}$$

where $\mu$ is the mean value of the Pareto distribution. This may be seen by neglecting the term $\lambda_{\mathrm{m}}^{k-1}$ (since typically $\lambda_{\mathrm{m}}^{k-1} \ll 1$ when $k \geq 2$), and noticing that the ratio $\mu/\beta_1$ is then approximately (approximately because of neglecting $\lambda_{\mathrm{m}}^{k-1}$) bounded by

$$1 \lessapprox \mu/\beta_1 \lessapprox \frac{2^{-\frac{1-\ln 2}{\ln 2}}}{\ln 2} \approx 1.062,$$

where the lower bound is attained at $k = 2$ and the upper bound is attained at $k = (1 - \ln 2)^{-1}$. The mean value $\mu$ may then be estimated by

$$\hat{\mu} = \frac{1 - \hat{\lambda}_1(\boldsymbol{m})}{2^{N_\mathrm{q}} - 1}. \tag{6.27}$$

For the $N > 2$ case, it is difficult to obtain closed-form solutions of $\boldsymbol{\beta}$. Furthermore, in general, the optimization problem with respect to $\boldsymbol{\beta}$ may no longer be convex. Fortunately, in the following proposition we show that $\tilde{\epsilon}(\boldsymbol{\beta})$ satisfies a generalized convexity property, which guarantees that the global optimum is always attainable.

**Proposition 6.1** (Invexity of the Permutation Filter Design Problem)**.** *The cost function $\tilde{\epsilon}(\boldsymbol{\beta})$ in* (6.17) *is an invex*[1] *function of $\boldsymbol{\beta}$ in the convex feasible region $\mathscr{B}$. In other words, every stationary point of $\tilde{\epsilon}(\boldsymbol{\beta})$ in $\mathscr{B}$ is a global minimum.*

*Proof.* Please refer to Appendix A.9. □

Proposition 6.1 implies that the following simple projected gradient descent iteration rule

$$\begin{aligned} \tilde{\boldsymbol{\beta}}^{(\ell+1)} &= \boldsymbol{\beta}^{(\ell)} - \delta^{(\ell)} \cdot \left.\frac{\partial \tilde{\epsilon}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right|_{\boldsymbol{\beta}^{(\ell)}}, \\ \boldsymbol{\beta}^{(\ell+1)} &= \mathcal{T}_{\mathscr{B}}\left[\tilde{\boldsymbol{\beta}}^{(\ell+1)}\right], \end{aligned} \tag{6.28}$$

may be used to solve the problem in (6.17), despite that $\tilde{\epsilon}(\boldsymbol{\beta})$ may not be convex with respect to $\boldsymbol{\beta}$. The operator $\mathcal{T}_{\mathscr{B}}(\cdot)$ projects its argument onto the convex feasible region $\mathscr{B}$, which can be implemented by simply sorting the entries of $\boldsymbol{\beta}$ after each iteration. The step size parameter $\delta^{(l)}$ can be determined using classic line search methods [193]. More sophisticated methods, such as modified Newton's method specifically tailored for invex optimization [194], may also be applied to accelerate the convergence.

According to our discussion in Appendix A.9, the cost function $\xi(\boldsymbol{\alpha})$ is a convex function of $\boldsymbol{\alpha}$. The reason that we do not solve directly this convex problem is that it is a challenge to differentiate the cost function $\xi(\boldsymbol{\alpha})$. By contrast, it is relatively simple to compute the gradient $\frac{\partial}{\partial \boldsymbol{\beta}}\tilde{\epsilon}(\boldsymbol{\beta})$, as follows:

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\beta}}\tilde{\epsilon}(\boldsymbol{\beta}) &= \sum_{i=0}^{N-1} (-1)^i \frac{\partial}{\partial \boldsymbol{\beta}} \int_{\beta_{N-i-1}}^{\beta_{N-i}} G_{\boldsymbol{\beta}}(\lambda)\mathrm{d}\lambda \\ &= \sum_{i=0}^{N-1} (-1)^i \int_{\beta_{N-i-1}}^{\beta_{N-i}} \frac{\partial}{\partial \boldsymbol{\beta}} G_{\boldsymbol{\beta}}(\lambda)\mathrm{d}\lambda \\ &= \sum_{i=0}^{N-1} (-1)^{i+1} \int_{\beta_{N-i-1}}^{\beta_{N-i}} g_{\boldsymbol{\beta}}(\lambda)\mathrm{d}\lambda, \end{aligned} \tag{6.29}$$

---

[1] Invexity is a generalization of convexity, ensuring that the global optimal solutions can be found by using the Karush-Kuhn-Tucker conditions [192].

**Algorithm 1** Type-2 permutation filter design

```
Input: Spectral density parameters $k$ and $\lambda_\mathrm{m}$
Output: The filter weight vector $\boldsymbol{\alpha}$
Initialisation: $\boldsymbol{\beta}^{(0)}$
$\ell = 0$;
repeat
    Compute $\frac{\partial \tilde{\epsilon}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\Big|_{\boldsymbol{\beta}^{(\ell)}}$ using (6.21), (6.22), (6.29) and (6.30);
    Determine $\delta^{(\ell)}$ using line search methods;
    Update $\boldsymbol{\beta}^{(\ell+1)} = \boldsymbol{\beta}^{(\ell)} - \delta^{(\ell)} \cdot \frac{\partial \tilde{\epsilon}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\Big|_{\boldsymbol{\beta}^{(\ell)}}$;
    Sort the entries in $\boldsymbol{\beta}^{(\ell+1)}$ in the ascending order;
    $\ell = \ell + 1$;
until convergence conditions are met
Compute $\boldsymbol{\alpha} = \boldsymbol{\varphi}(\boldsymbol{\beta}^{(\ell)})$ using (6.13);
return $\boldsymbol{\alpha}$
```

where $[\boldsymbol{g}_{\boldsymbol{\beta}}(\lambda)]_i = \lambda^{-k} \prod_{\substack{n=1 \\ n \neq i}}^{N-1} (\lambda - \beta_n)$. The order between the integration and the differentiation is interchangeable, since $G_{\boldsymbol{\beta}}(\lambda) = 0$ for $\lambda = \beta_i$, $\forall i = 1, 2, \ldots, N-1$. The integrals can be computed using (6.21) and (6.22), but for $[\boldsymbol{g}_{\boldsymbol{\beta}}(\lambda)]_i$ the vector $\boldsymbol{\alpha}$ should be replaced by

$$\tilde{\boldsymbol{\alpha}}_i = \mathop{\bigstar}_{n=1, n \neq i}^{N-1} [1 \ \ -\beta_n]^{\mathrm{T}}. \tag{6.30}$$

When low-complexity methods are preferred, a simple heuristic alternative, which will be referred to as the "Type-1 permutation filter", is to set

$$\beta_1 = \beta_2 = \ldots = \beta_{N-1} = \mu. \tag{6.31}$$

Correspondingly, we refer to the aforementioned optimization-based method, summarized in Algorithm 1, as the "Type-2 permutation filter". In Section 6.3 we will show that, even though the Type-1 filters rely on a heuristic method, they are capable of outperforming VD.

To conclude, the complete workflow of an $N$-th order permutation filter for a given observable $\mathcal{U}$ consists of the following steps:

1. Execute the original circuit and obtain the estimate of $\mathrm{Tr}\,\{\rho\mathcal{U}\}$;
2. Execute all $n$-th order virtual distillation circuits ($2 \leq n \leq N$), and obtain the estimates of $\mathrm{Tr}\,\{\rho^n\mathcal{U}\}$ as well as additional observations $m_{n-1} = \mathrm{Tr}\,\{\rho^n\}$;
3. Fit the spectral density model using the observations $\boldsymbol{m} = [m_1 \ \ldots \ m_{N-1}]^{\mathrm{T}}$, and determine the filter parameters $\boldsymbol{\alpha}$;
4. Obtain the final filtered result by classical post-processing.

### 6.2.3 The Computational Overhead of Permutation Filters

In terms of the number of gates, the computational overhead of permutation filters is the same as virtual distillation. The number of gates in order to implement the permutation operation $\mathcal{P}_n$ (which is the additional gate cost of the protocol compared to the unprotected circuit) has been discussed in [65]. Specifically, if the original unprotected circuit acts on $N_\mathrm{q}$ qubits, implementing $\mathcal{P}_n$ would require $N_\mathrm{q}(n-1)$ controlled-SWAP gates (i.e. the Fredkin gate), which is on the order of $O(N_\mathrm{q})$. Hence we may conclude that the method would be beneficial when the algorithm circuit has an increasing depth with respected to $N_\mathrm{q}$.

As for the sampling overhead, permutation filters are slightly different from virtual distillation due to the weighted averaging process. For virtual distillation, an approximate expression for the variance of a given observable $\mathcal{U}$ has been presented in [66]. Using similar arguments, we may also obtain an expression for permutation filters as

$$\begin{aligned}
&\mathrm{Var}\{y_{\mathrm{filter}}^{(N)}(\mathcal{U})\} \\
&\approx \frac{1-\sum_{n=1}^{N}\alpha_{N-n+1}^{2}\mathrm{Tr}\left\{\rho^{n}\mathcal{U}\right\}^{2}}{(\alpha_N+\sum_{n=2}^{N}\alpha_{N-n+1}\mathrm{Tr}\left\{\rho^{n}\right\})^{2}} \\
&-\frac{2(\sum_{n=1}^{N}\alpha_{N-n+1}\mathrm{Tr}\left\{\rho^{n}\mathcal{U}\right\})}{(\alpha_N+\sum_{n=2}^{N}\alpha_{N-n+1}\mathrm{Tr}\left\{\rho^{n}\right\})^{3}} \\
&\times\sum_{n=2}^{N}\alpha_{N-n+1}^{2}\left(\mathrm{Tr}\left\{\rho\mathcal{U}\right\}-\mathrm{Tr}\left\{\rho^{n}\mathcal{U}\right\}\mathrm{Tr}\left\{\rho^{n}\right\}\right) \\
&+\frac{(\sum_{n=1}^{N}\alpha_{N-n+1}\mathrm{Tr}\left\{\rho^{n}\mathcal{U}\right\})^{2}(1-\sum_{n=2}^{N}\alpha_{N-n+1}^{2}\mathrm{Tr}\left\{\rho^{n}\right\}^{2})}{(\alpha_N+\sum_{n=2}^{N}\alpha_{N-n+1}\mathrm{Tr}\left\{\rho^{n}\right\})^{4}}.
\end{aligned} \tag{6.32}$$

The variance of the entire Hamiltonian $\mathcal{H}$ can then be calculated by a weighted summation over the Pauli observables. In light of this, the sampling overhead factor of permutation filters may be defined as the ratio between the variance of the Hamiltonian estimator based on the permutation filter and that based on the unprotected circuit. We will evaluate the sampling overhead of permutation filters applied to practical variational quantum algorithms using this metric in Section 6.4.3.

## 6.3 The Error Reduction Performance of Permutation Filters

In this section, we quantify the error reduction of permutation filters compared to VD of the same order using the following performance metric.

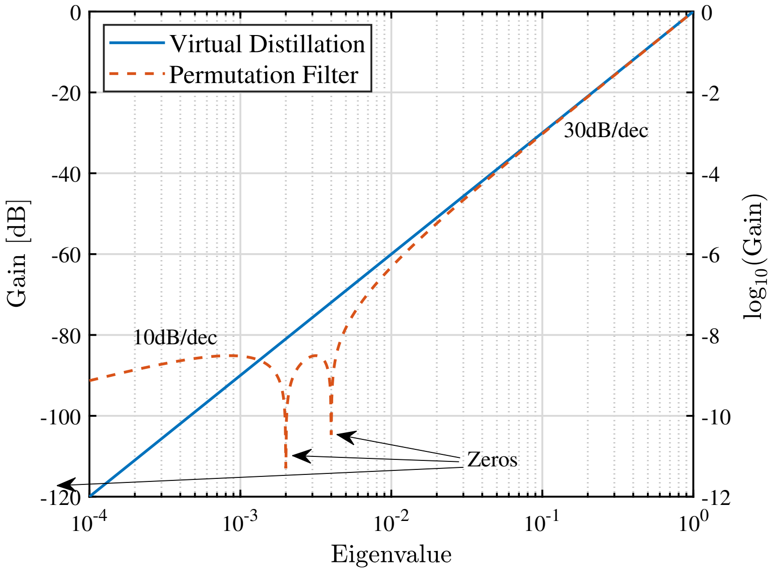


FIGURE 6.8: The spectral response of a third-order permutation filter, compared to that of the third-order VD.

**Definition 6.2** (Error Ratio)**.** We define the error ratio between an $N$-th order permutation filter $\mathcal{F}_{\boldsymbol{\beta}}(\cdot)$ and its corresponding $N$-th order counterpart based on VD as follows:

$$R(\boldsymbol{\beta}) := \frac{\tilde{\epsilon}(\boldsymbol{\beta})}{\tilde{\epsilon}(\mathbf{0})}. \tag{6.33}$$

Note that VD is equivalent to a permutation filter that satisfies $\boldsymbol{\beta} = \mathbf{0}$.

Intuitively, the permutation filters are narrowband notch filters, hence they should perform better when the "bandwidth" of the undesired spectral components is lower. To see this more clearly, we consider the spectral response of a third-order permutation filter, as portrayed in Fig. 6.8. Observe that every zero contributes 10 dB per decade to the slope of the filter gain.[2] For both third-order permutation filters and for VD, the slope will be 30 dB per decade beyond the largest zero. In light of this, the only region where permutation filters have smaller gain is the narrowband range around the two largest zeros. Therefore, permutation filters perform the best when the noise components are concentrated in this region.

To make our aforementioned intuitions more rigorous, we define the following quantities to characterize the bandwidth.

**Definition 6.3** (Noise Bandwidth)**.** We define the bandwidth of the noise (i.e., the undesired spectral components $\tilde{\boldsymbol{\lambda}}$ in a mixed state $\rho$) as follows:

$$B(\tilde{\boldsymbol{\lambda}}) := \sqrt{\mathbb{E}\{|\lambda - \mu|^2\}}, \tag{6.34}$$

---

[2] For readers do not familiar with classical signal processing theory, please refer to Appendix B.2 for further explanation.

where

$$\mathbb{E}\{g(\lambda)\} := \int_{\lambda_{\mathrm{m}}}^{1} g(\lambda) f(\lambda) \mathrm{d}\lambda, \tag{6.35}$$

denotes the expectation operation, and $\mu = \mathbb{E}\{\lambda\}$ denotes the mean value of noise components. We also define the *relative noise bandwidth* as

$$b(\tilde{\boldsymbol{\lambda}}) := \mu^{-1} B(\tilde{\boldsymbol{\lambda}}). \tag{6.36}$$

Given the previous definitions, we are now prepared to state the following result concerning the error ratio of Type-1 permutation filters.

**Proposition 6.4** (Generic Error Ratio Scaling Behaviour of Type-1 Permutation Filters)**.** *The error ratio $R(\boldsymbol{\beta})$ of an $N$-th order Type-1 permutation filter, as a function of the relative noise bandwidth $b(\tilde{\boldsymbol{\lambda}})$, can be bounded by*

$$R(\boldsymbol{\beta}) \leq \frac{1}{\mu} \left[ b(\tilde{\boldsymbol{\lambda}}) \sqrt{2^{N_{\mathrm{q}}} - 1} \right]^{N-1}, \tag{6.37}$$

*as $b(\tilde{\boldsymbol{\lambda}}) \to 0$.*

*Proof.* The term $\tilde{\epsilon}(\mathbf{0})$ can thus be written explicitly as

$$\tilde{\epsilon}(\mathbf{0}) = \mathbb{E}\{\lambda^N\}. \tag{6.38}$$

Using Jensen's inequality [195], we have

$$\tilde{\epsilon}(\mathbf{0}) \geq [\mathbb{E}\{\lambda\}]^N = \mu^N. \tag{6.39}$$

Therefore, from (6.33) we obtain

$$\begin{aligned} R(\boldsymbol{\beta}) &\leq \tilde{\epsilon}(\boldsymbol{\beta}) \mu^{-N} \\ &= \mathbb{E}\{|\lambda(\lambda - \mu)^{N-1}|\} \mu^{-N} \\ &\leq \frac{1}{\mu} \cdot \mathbb{E}\left\{ \left| (\lambda - \mu) \mu^{-1} \right|^{N-1} \right\}, \end{aligned} \tag{6.40}$$

where the last line follows from the fact that $\lambda \leq 1$ holds for all eigenvalues. Furthermore, assume that we have access to the actual values of $\tilde{\boldsymbol{\lambda}}$ (which will only be used for calculating intermediate results), we have

$$
\begin{aligned}
\mathbb{E}\left\{\left|\frac{\lambda-\mu}{\mu}\right|^{N-1}\right\} &= \left(\frac{\mu^{-1}\left\|\tilde{\boldsymbol{\lambda}}-\mu\mathbf{1}\right\|_{N-1}}{(2^{N_{\mathrm{q}}}-1)^{\frac{1}{N-1}}}\right)^{N-1} \\
&\le \left(\mu^{-1}\left\|\tilde{\boldsymbol{\lambda}}-\mu\mathbf{1}\right\|_{\infty}\right)^{N-1} \\
&\le \left(\mu^{-1}\left\|\tilde{\boldsymbol{\lambda}}-\mu\mathbf{1}\right\|_{2}\right)^{N-1} \\
&= \left(b(\tilde{\boldsymbol{\lambda}})\sqrt{2^{N_{\mathrm{q}}}-1}\right)^{N-1}.
\end{aligned}
\tag{6.41}
$$

Hence the proof is completed. □

Proposition 6.4 supports our intuition that the error ratio decreases, as the noise bandwidth becomes smaller. However, the constant $\sqrt{2^{N_{\mathrm{q}}}-1}$ in (6.37) can be extremely large for large $N_{\mathrm{q}}$, when the bound becomes of limited practical significance. In the following result we show that for spectral densities satisfying Pareto distributions, the dependence of the bound on $N_{\mathrm{q}}$ can be eliminated.

**Proposition 6.5** (Type-1 Filters Applied to Pareto-Distributed States)**.** *Assume that $f(\lambda)$ corresponds to a Pareto distribution, and that $b(\tilde{\boldsymbol{\lambda}}) < (N-1)^{-1}$. The error ratio of an N-th order Type-1 permutation filter can be bounded by*

$$
\begin{aligned}
R(\boldsymbol{\beta}) &\le \frac{(N-1)![1+b(\tilde{\boldsymbol{\lambda}})]^{N}}{e\prod_{n=1}^{N-2}[1-nb(\tilde{\boldsymbol{\lambda}})]}\cdot[b(\tilde{\boldsymbol{\lambda}})]^{N-1} \\
&= O\left\{[b(\tilde{\boldsymbol{\lambda}})]^{N-1}\right\}.
\end{aligned}
\tag{6.42}
$$

*Proof.* Please refer to Appendix A.10. □

Both Proposition 6.4 and 6.5 show that, the error ratio of Type-1 filters decreases exponentially with the filter order $N$. For Type-2 filters, this may be viewed as an upper bound of the error ratio, since their parameter vectors $\boldsymbol{\beta}$ are obtained via optimization. By contrast, the parameter vectors of Type-1 filters are determined using only the mean value of noise components, hence are suboptimal.

A natural question that arises is: under what practical conditions do the undesired spectral components have small relative bandwidth? In the following proposition, we show that the relative noise bandwidth decreases with the depth of quantum circuits, as well as with the error rate of the gates in the circuits.

**Proposition 6.6** (Exponential Spectral Concentration of Deep Quantum Circuits)**.** *Assume that each qubit is acted upon by at least L gates, and that each of the gates is contaminated by quantum channels containing Pauli noise, which have matrix representations under the Pauli basis given in* (A.108)*. We assume furthermore that the probability of each type of*

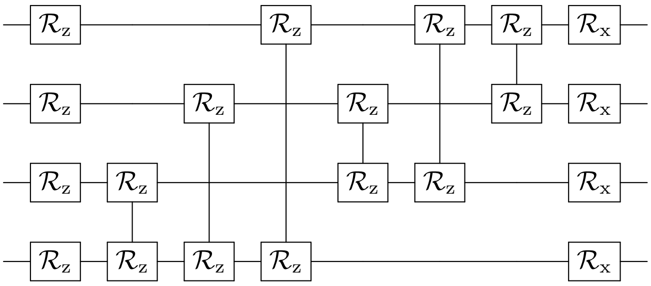


FIGURE 6.9: Schematic of a stage in the parametric state preparation circuit used in the simulations. Here we set $N_{\mathrm{q}} = 4$ only for illustration.

*Pauli error (i.e., X error, Y error or Z error) on each qubit is lower bounded by* $\epsilon_{\mathrm{l}}$*. Under these assumptions, the relative noise bandwidth can be upper bounded by*

$$\begin{aligned} b(\tilde{\boldsymbol{\lambda}}) &\leq \frac{1+\sqrt{2^{N_{\mathrm{q}}}-1}}{1-2^{-N_{\mathrm{q}}}-\exp(-4\epsilon_{\mathrm{l}}L)} \cdot \exp(-4\epsilon_{\mathrm{l}}L) \\ &= O\left\{\exp(-4\epsilon_{\mathrm{l}}L)\right\}. \end{aligned} \tag{6.43}$$

*Proof.* Please refer to Appendix A.11. □

From Proposition 6.6 we observe that the relative noise bandwidth decreases exponentially with the product of $\epsilon_{\mathrm{l}}$ and $L$. This implies that the proposed permutation filters would provide more significant performance improvements when the circuits are relatively deep, or the gates therein are noisy.

## 6.4 Numerical Results

In this section, we further illustrate the results discussed in the previous sections using numerical simulations. In all simulations, we consider a class of parametric state preparation circuit consisting of different number of stages, for which a single stage is portrayed in Fig. 6.9. For illustration we drawn a four-qubit circuit, but in the actual simulations we set $N_{\mathrm{q}} = 10$. As observed from Fig. 6.9,[3] each stage of the circuit is constructed by two-qubit ZZ-rotation gates acting upon each pair of qubits, and single-qubit X- and Y-rotation gates acting upon each qubit. The rotation angle of each gate is a parameter to be determined. In the simulations, we choose the parameters by independent sampling from uniform distributions over $[-\pi, \pi]$, and the simulation results are averaged over 100 random instances of the circuits. The gates are inflicted by depolarizing errors occurring at varying probabilities, but we always set the depolarizing probabilities of two-qubit gates 10 times higher than that of single-qubit gates.

---

[3] The notation Rx, Ry, and Rz denote X-, Y-, and Z-rotation gates, respectively.

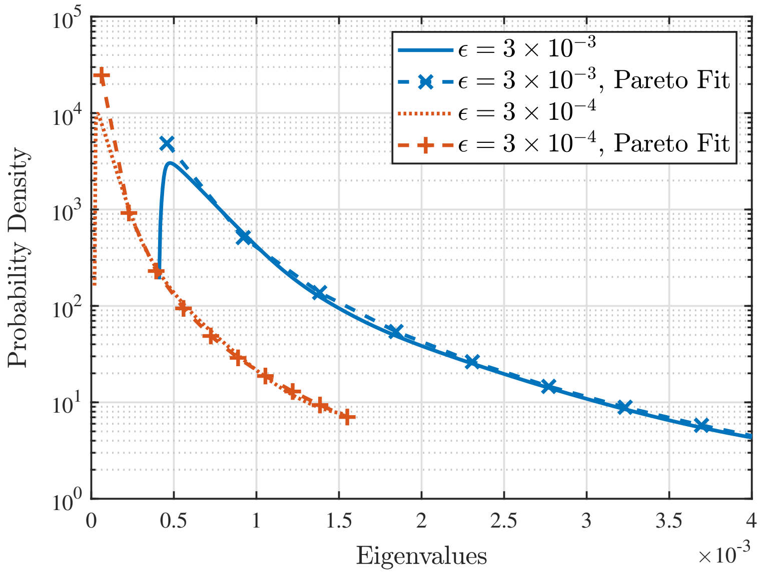


FIGURE 6.10: The spectra of the output states of parametric state preparation circuits having different depolarizing probability $\epsilon$.

### 6.4.1 Spectral Properties of the Output States

We first demonstrate the spectral densities of the output states. In particular, we consider parametric state preparation circuits having 10 stages acting on $N_\mathrm{q} = 10$ qubits. The spectral densities and the corresponding cumulative density functions for $\epsilon = 3 \times 10^{-4}$ and $\epsilon = 3 \times 10^{-3}$ are portrayed in Fig. 6.10 and Fig. 6.11, where $\epsilon$ denotes the depolarizing probability of each two-qubit gate. The Pareto fit are also plotted for comparison. We see that the Pareto distributions provide good approximations to the eigenvalue spectra, except for very small eigenvalues. This also suggests that the Pareto fit may become less accurate when the noise bandwidth is very narrow, for which the approximation error becomes more significant.

### 6.4.2 The Filter Design Metric

Next, we investigate the values of the cost function $\tilde{e}(\boldsymbol{\beta})$ for filter design under different scenarios, which may be used for evaluating the performance of the filters irrespective of the specific choices of observables.

In Fig. 6.12, we compare the values of $\tilde{e}(\boldsymbol{\beta})$ obtained both by our permutation filters and by VD, as functions of the number of stages in the state preparation circuits. The depolarizing probability of two-qubit gates is $1.25 \times 10^{-3}$. In this figure, the curve "Closed-form, 2nd order" corresponds to the second-order permutation filter designed based on the closed-form solution in (6.25)–(6.27). We observe from the figure that permutation filters significantly outperform VD, when the number of stages is large,

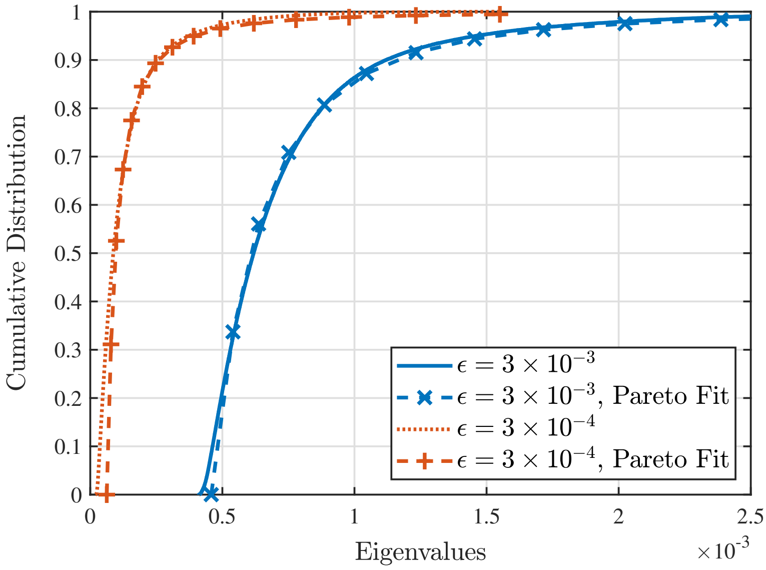


FIGURE 6.11: The Pareto fits of the output spectra of parametric state preparation circuits having different depolarizing probability $\epsilon$.

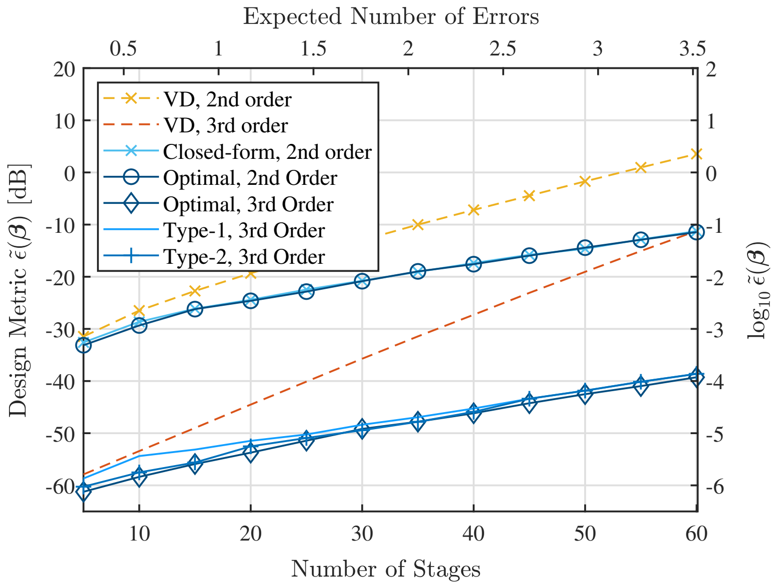


FIGURE 6.12: The value of the design metric $\tilde{\epsilon}(\boldsymbol{\beta})$ in (6.19) for both VD and for the proposed methods, as functions of the number of stages. The two-qubit depolarizing probability is $1.25 \times 10^{-3}$.

for both the second-order case and the third-order case. In particular, in the second-order case, both the Type-1 and Type-2 permutation filters have the same parameters $\boldsymbol{\beta}$, and we see that their performance is very close to that of the optimal solution, which is obtained by directly solving (6.17) relying on the full *a priori* knowledge of the spectral density $f(\lambda)$.

For the third-order case, we see that the Type-2 filter slightly outperforms the Type-1 filter, when the number of stages is relatively small. Intuitively, by adjusting the two

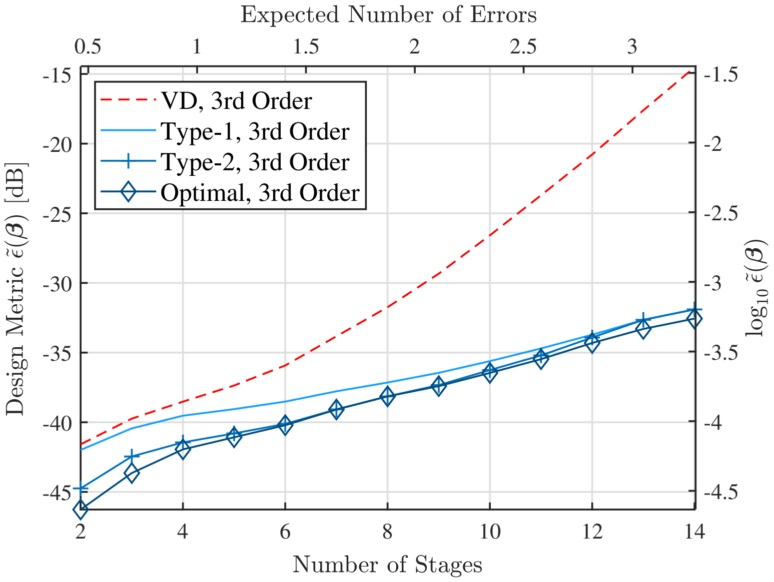


FIGURE 6.13: The value of the design metric $\tilde{\epsilon}(\beta)$ in (6.19) for both VD and for the proposed methods, as functions of the number of stages. The two-qubit depolarizing probability is $5 \times 10^{-3}$.

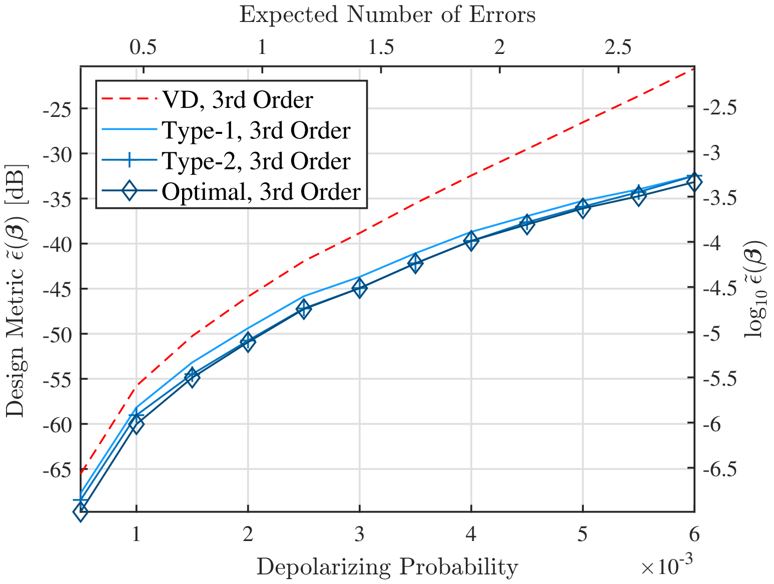


FIGURE 6.14: The value of the design metric $\tilde{\epsilon}(\beta)$ in (6.19) for both VD and for the proposed methods, as functions of the depolarizing probability.

zeros of the third-order filters, it is indeed possible to achieve a better error-reduction performance than that of simply placing the zeros at the same point. However, the effect of adjusting the positions of zeros would be less significant when the noise bandwidth is smaller, corresponding to the case where the number of stages is large. Closer scrutiny reveals that the performance of both the Type-1 and Type-2 third-order filters become similar when the number of stages is large, especially when it is larger than 45. By contrast, the performance of the Type-2 filter is near-optimal when the number of stages is moderate (around 25-40). This trend may prevail, because the Pareto fit

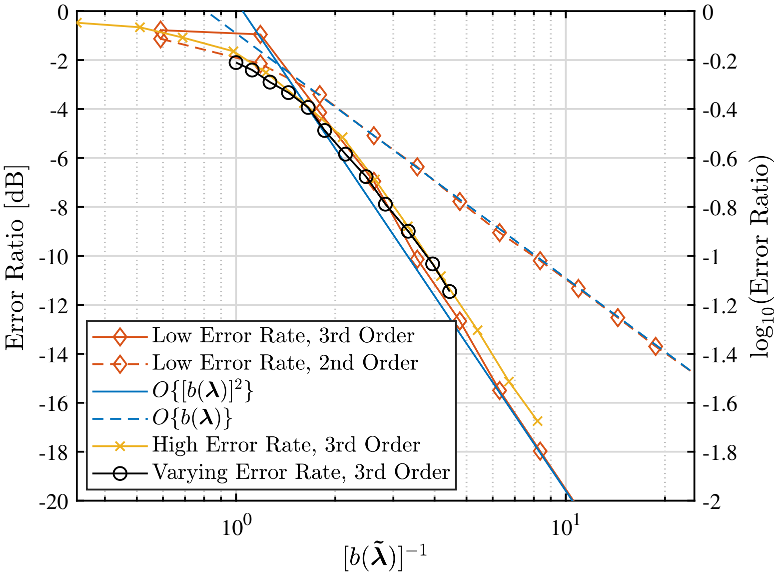


FIGURE 6.15: The error ratio $R(\boldsymbol{\beta})$ in (6.33) between Type-1 permutation filters and VD vs. the reciprocal of the relative noise bandwidth $b(\tilde{\boldsymbol{\lambda}})$.

becomes more accurate, when the noise bandwidth is moderate.

In Fig. 6.13, we consider the case where the two-qubit depolarizing probability is $5 \times 10^{-3}$, which is four times that of Fig. 6.12. The trends of the curves are similar to those of the lower depolarizing probability scenario. It may now be seen more clearly that the Type-2 permutation filter substantially outperforms its Type-1 counterpart, when the number of stages is small.

Next, in Fig. 6.14, we consider circuits having varying depolarizing probabilities. The number of stages is fixed to 10. We observe a similar increasing gap between the permutation filters and VD. In addition, the Type-2 permutation filter also exhibits better performance for moderate depolarizing probabilities.

In Fig. 6.15, we illustrate the scaling behaviour of the error ratio between the type-1 permutation filters and VD, which has been discussed in Section 6.3. In particular, we plot the error ratios computed using the data presented in Figures 6.12, 6.13 and 6.14. We observe that when the relative noise bandwidth $b(\tilde{\boldsymbol{\lambda}})$ is small (less than around 0.5), all error ratios are reduced roughly polynomially with $[b(\tilde{\boldsymbol{\lambda}})]^{-1}$. Furthermore, the slopes of the curves are almost equal to the asymptotes scaling quadratically and linearly with $b(\tilde{\boldsymbol{\lambda}})$, respectively for third-order and second-order filters. These observations corroborate Propositions 6.4 and 6.5.

Finally, in Fig. 6.16 and Fig. 6.17, we demonstrate that the commonly used metric of noisiness, namely the expected number of errors, does not determine the relative noise bandwidth on its own, and hence does not solely determine the error ratio between permutation filters and VD. To this end, we fixed the number of expected errors, and change the number of stages and the depolarizing probability accordingly. As it can

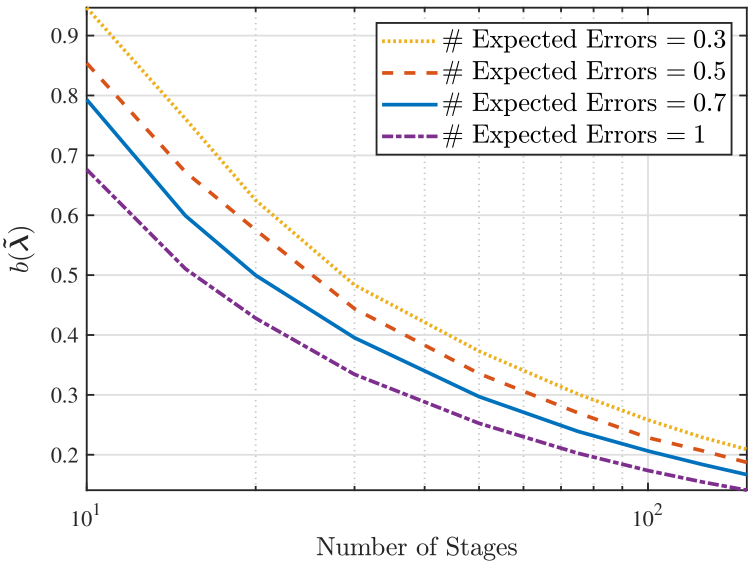


FIGURE 6.16: The relative noise bandwidth $b(\tilde{\boldsymbol{\lambda}})$.

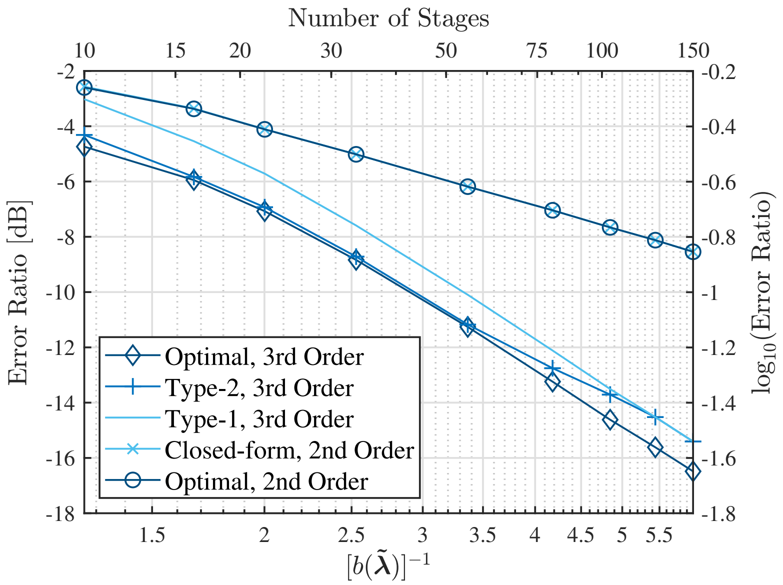


FIGURE 6.17: The error ratio $R(\boldsymbol{\beta})$ in (6.33) between permutation filters and VD. The number of expected errors is 0.7.

be seen from Fig. 6.16, the relative bandwidth shrinks with the number of stages, even when the number of expected errors is fixed. Similarly, we observe from Fig. 6.16 and Fig. 6.17 that the error ratio decreases with decreasing depolarizing probability (or increasing number of stages).

From the discussions in this subsection, we may conclude that the benefit of the permutation filter method is more significant when the circuit is rather noisy, or it is deep but is constituted by gates having relatively small error probabilities.

### 6.4.3 Case Study: QAOA-Aided Multi-User Detection

In this subsection we demonstrate the performance of permutation filters when applied to a practical variational quantum algorithm, namely the QAOA. The parametric state-preparation circuits of QAOA are multi-stage circuits having an alternating structure, which take a plus state $|+\rangle^{\otimes N_{\mathrm{q}}}$ as the input and produce the following output

$$|\psi\rangle_{\mathrm{out}} = e^{-\imath b_{N_{\mathrm{L}}} \mathcal{H}_{\mathrm{M}}} e^{-\imath c_{N_{\mathrm{L}}} \mathcal{H}_{\mathrm{P}}} \dots e^{-\imath b_1 \mathcal{H}_{\mathrm{M}}} e^{-\imath c_1 \mathcal{H}_{\mathrm{P}}} |+\rangle^{\otimes N_{\mathrm{q}}}, \tag{6.44}$$

where $N_{\mathrm{L}}$ denotes the number of stages, $\mathcal{H}_{\mathrm{M}}$ denotes the mixing Hamiltonian defined as $\mathcal{H}_{\mathrm{M}} := \sum_{n=1}^{N_{\mathrm{q}}} \mathcal{X}_i$ ($\mathcal{X}_i$ denotes the Pauli-X operator acting on the $i$-th qubit), and $\mathcal{H}_{\mathrm{P}}$ denotes the phase Hamiltonian that encodes the problem to be solved. The parameters $\boldsymbol{b} = [b_1, \dots, b_{N_{\mathrm{L}}}]^{\mathrm{T}}$ and $\boldsymbol{c} = [c_1, \dots, c_{N_{\mathrm{L}}}]^{\mathrm{T}}$ control the dynamic of the algorithm, and are typically determined by an iterative optimization procedure [32]. Since we focus on the performance evaluation for error mitigation methods, here we consider a suboptimal linear scheduling [185] instead of optimizing for the parameters, given by $c_\ell = \ell / N_{\mathrm{L}}$ and $b_\ell = 1 - \ell / N_{\mathrm{L}}$.

In particular, we construct the phase Hamiltonian corresponding to the multi-user detection problem [186] for wireless communication systems.[4] For an $m \times n$ multiple-input mutliple-output (MIMO) system, the received signal may be modelled as

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{\omega},$$

where $\boldsymbol{H}$ denotes the MIMO channel, $\boldsymbol{x}$ represents the transmitted signal, and $\boldsymbol{\omega}$ denotes the noise. For simplicity of the illustration, we assume that the noise is i.i.d. Gaussian on each receiving antenna, and that the modulation scheme is binary phase-shift keying (BPSK), hence $\boldsymbol{x} \in \{-1, 1\}^n$ and $\boldsymbol{H} \in \mathbb{R}^{m \times n}$. The phase Hamiltonian corresponding to the maximum likelihood estimator of $\boldsymbol{x}$ is thus given by

$$\sum_{k=1}^{n} [\boldsymbol{H}^{\mathrm{T}}\boldsymbol{y}]_i \mathcal{Z}_i - \sum_{i=1}^{n-1} \sum_{j>i} [\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}]_{i,j} \mathcal{Z}_i \mathcal{Z}_j. \tag{6.45}$$

We consider the following scenario for the numerical simulation: $N_{\mathrm{q}} = m = n = 10$, the channel $\boldsymbol{H}$ has i.i.d Gaussian entries with zero mean and a variance of $1/m = 0.1$, and the signal-to-noise ratio is 13dB, implying that $[\boldsymbol{\omega}]_i \sim \mathcal{N}(0, 0.05)$.

We first fix the number of expected errors at 0.7 and investigate the dependency of the computational error (the absolute difference between the error-free result and the result computed relying on noisy circuits based on the entire Hamiltonian) on the number of stages. As it may be seen from Fig. 6.18, the permutation filters are more beneficial when the circuit is deep, as have been discussed in Section 6.4.2. We may also observe

---

[4] For readers not familiar with wireless communication, just note that it is a quadratic unconstrained binary optimization problem.

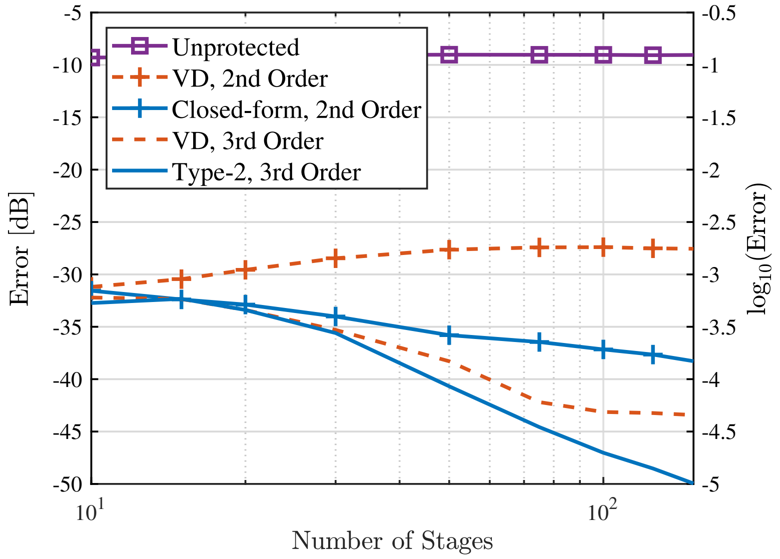


FIGURE 6.18: The computational error of permutation filters applied to QAOA-aided multi-user detection vs. the number of stages, where the number of expected errors is fixed at 0.7.

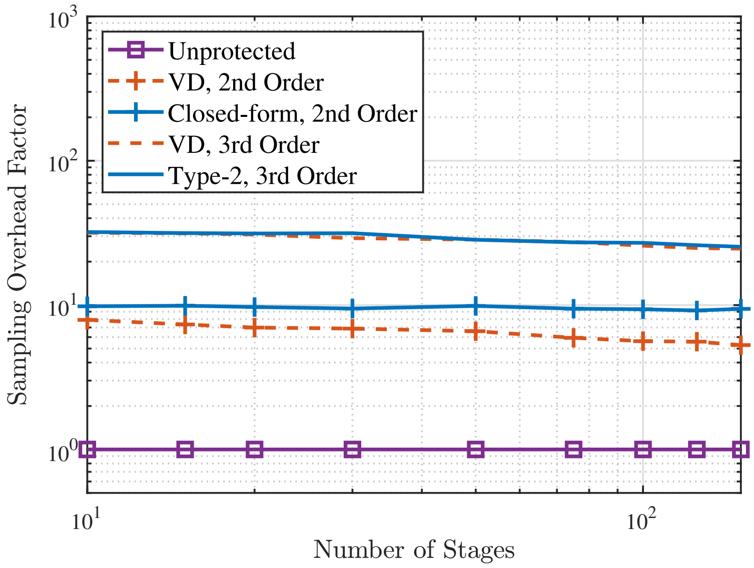


FIGURE 6.19: The sampling overhead factor of permutation filters applied to QAOA-aided multi-user detection vs. the number of stages, where the number of expected errors is fixed at 0.7.

from Fig. 6.19 that the sampling overhead is nearly constant with the number of stages, suggesting that the number of expected errors might be the principal determining factor of the overhead.

Next, we present the relationship between the computational error and the number of expected error, with a fixed number of stages $N_{\mathrm{L}} = 50$, in Fig. 6.20. It is seen from the figure that the permutation filter improves the error mitigation performance significantly when the number of expected errors is large. However, it should also be

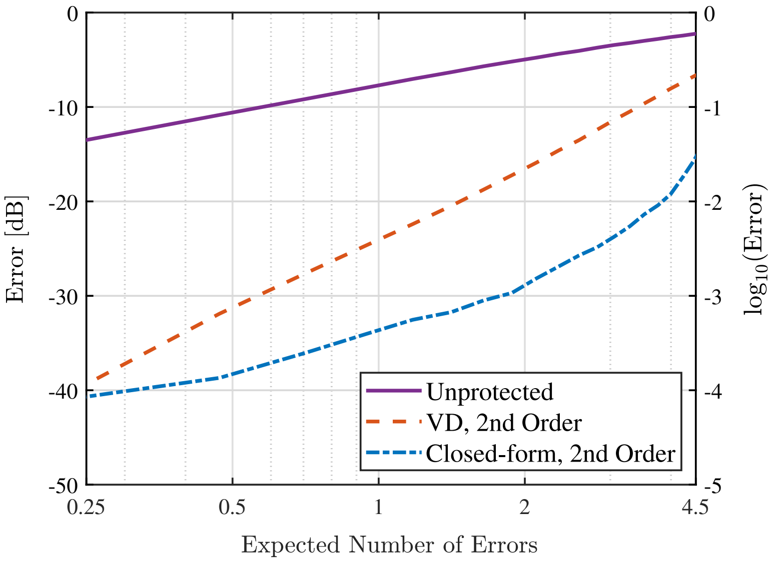


FIGURE 6.20: The computational error of permutation filters applied to QAOA-aided multi-user detection vs. the number of expected errors, where the number of stages is fixed at 50.

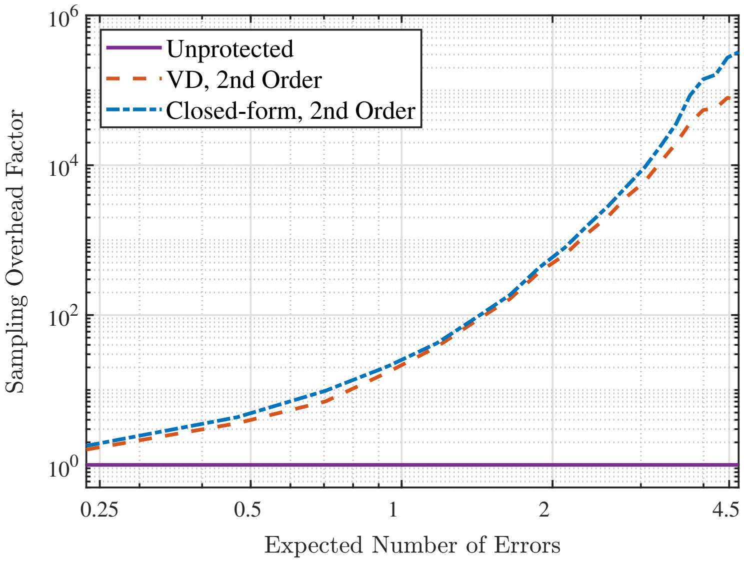


FIGURE 6.21: The sampling overhead factor of permutation filters applied to QAOA-aided multi-user detection vs. the number of expected errors, where the number of stages is fixed at 50.

noted that the sampling overhead increases dramatically when the number of expected errors is larger than 1, as shown in Fig. 6.21. Extra care should be taken for this issue, since a high sampling overhead may render the error mitigation method unfavorable in practice.

## 6.5 Conclusions and Future Research

### 6.5.1 Conclusions

In this chapter, we have proposed a general framework for designing FIR-like permutation filters for mitigating the computational errors of variational quantum algorithms. In particular, the filter takes a polynomial form, combining different order of VD circuits. The original VD circuits may be viewed as special cases of permutation filters having only one non-zero coefficient. We have also shown that the filter design problem is an invex problem, hence the algorithm is guaranteed to converge to the global optimum. For narrowband noise scenarios, we have also shown a polynomial error reduction compared to VD. This implies that permutation filters improve the error-reduction performance more substantially for quantum circuits having large depth or higher gate error rate.

### 6.5.2 Future Research

- We note that the performance metric we used for filter design is an upper bound of the error magnitude across all unitary observables. A possible future research direction is to find other metrics better suited to specific classes of practical observables. For example, many combinatorial optimization problems and quantum chemistry problems may be encoded in Hamiltonians that can be decomposed into a weighted sum of low-degree Pauli strings [62]. One may derive tighter upper bounds of the error magnitude particularly for these Pauli strings.
- From a broader perspective, permutation filter is a technique aiming for extracting the dominant eigenvector from a noisy quantum state represented by its density matrix. In this context, a related technique is quantum principal component analysis (QPCA) [196], which extracts the principal components (i.e., the eigenvectors associated with the largest $k$ eigenvalues) by using fractional-order swap operators and quantum phase estimation. Specifically, in virtual distillation and permutation filters we use the original swap operator $\mathcal{S}$ between different copies of the state $\rho$ to obtain an integer power of $\rho$, such as $\rho^2$ and $\rho^3$. By contrast, in QPCA one use the fractional-order swap operator $e^{jt\mathcal{S}}$ to implement the Hamiltonian simulation taking the density matrix $\rho$ as the Hamiltonian, which is obtained using the following approximation:

$$\begin{aligned}\operatorname{tr}_\rho\left\{e^{-j\mathcal{S}t}(\rho\otimes\sigma)e^{j\mathcal{S}t}\right\} &= \cos^2 t\sigma + \sin^2 t\rho - j\sin t\cos t[\rho,\sigma] \\ &= \sigma - jt[\rho,\sigma],\end{aligned} \tag{6.46}$$

which corresponds to the evolution of $e^{-jt\rho}\sigma e^{jt\rho}$. This enables us to estimate the eigenvalues (especially, the largest eigenvalue) of $\rho$ using quantum phase estimation, as discussed in Section 2.5.1. Furthermore, since we focus on the dominant eigenvector, we may design a filter directly in the spectral domain to extract the largest eigenvalue and uncompute the phase estimation subroutine, following as similar design philosophy as the HHL algorithm [16] discussed in Section 2.5.1. This procedure is portrayed in Fig. 6.22.

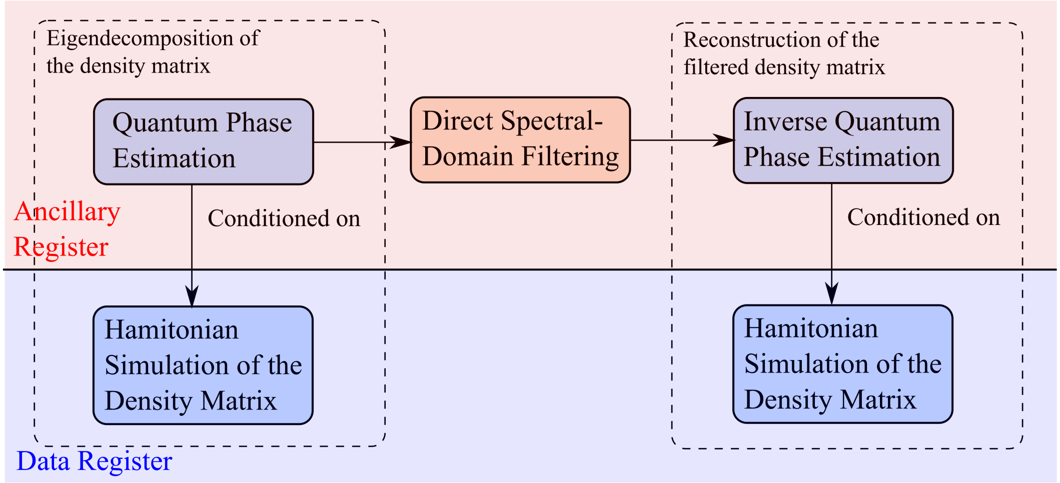


FIGURE 6.22: The workflow of the direct spectral-domain filter for permutation symmetry based QEM.

Compared to permutation filters, the QPCA-based method might has a lower sampling overhead, since it applies the filter directly on the spectral domain, and do not need to compute higher-order quantities such as $\rho^3$ that require a large number of samples. However, this potential benefit comes at the price of more sophisticated quantum circuits due to the quantum phase estimation subroutine, which may impose additional errors on the computational results. Therefore, an interesting research topic would be striking a beneficial trade-off between circuit depth and sampling overhead by amalgamating QPCA with permutation filters.

# Chapter 7

# Circuit Symmetry Based QEM

We have established in Chapter 6 that virtual distillation, as a special case of symmetry verification concerning permutation symmetry, can be generalized to a class of FIR-like filters of quantum states, namely the permutation filters. They may be viewed as "spatial filters" verifying the spatial consistency of quantum circuits, since qubits are typically physical entities that occupy disjoint spatial areas. In this chapter, we present another possible direction of generalizing symmetry verification, by exploiting circuit symmetries instead of state symmetries. This enables us to verify the temporal consistency of quantum circuits.

A related topic, namely the superposition of causal orders [76–78], which can be physically realized using the quantum switch of [79], has been investigated from the perspective of quantum communication. Specifically, it has been shown that the capacity of two quantum channels $\mathcal{A}$ and $\mathcal{B}$ may be improved by producing a coherent superposition between their compositions of different orders, i.e. $\mathcal{A} \circ \mathcal{B}$ and $\mathcal{B} \circ \mathcal{A}$ [197–199]. More surprisingly, non-zero capacity is achievable even if both the capacity of $\mathcal{A}$ and that of $\mathcal{B}$ are zero [200]. The implementation of the quantum switch relies on a control qubit, the state of which may be used to indicate the commutativity between the composite channels.

In this chapter, we argue that the quantum switch based method can be beneficially used for QEM, with some modifications. In particular, the quantum switch and its derivations are capable of verifying circuit symmetries such as the commutativity between quantum gates. This is in stark contrast to existing symmetry verification methods relying on stabilizer checks, which aim for verifying the specific properties of quantum states instead of circuits.

To be more specific, for quantum circuits consisting of mutually commuting gates, the original form of the quantum switch to verify the gate commutativity. However, the gate commutativity is a relatively weak symmetry condition, since it cannot be used to identify any error that commute with the gate itself. To this end, for quantum circuits that are known to commute with certain operators, especially Pauli operators, we propose a modified quantum switch based method termed as the "spatio-temporal stabilizer (STS)", which may be used for detecting and mitigating errors violating the commutativity condition. In contrast to conventional stabilizer-based symmetry verification, STSs do not depend on the specific quantum state, hence they are more generally applicable.

In Section 3.3.2, we have mentioned a fundamental difference between state symmetries and circuit symmetries, namely the simultaneous observability. This property is important, because only those symmetry conditions that can be simultaneously verified can contribute to the error mitigation capability at the same time. In particular, state symmetries are by definition simultaneously observable, since they form a stabilizer group. However, as will be shown in Section 7.3.3 this does not necessarily apply to circuit symmetries, hence the conditions under which STSs are simultaneously observable becomes practically significant.

We organize the rest of this chapter as follows. In Section 7.1, we elaborate on the difference between state symmetries and circuit symmetries. Then, in Section 7.2, we present the implementations of the quantum switch for verifying gate commutativity. For circuits having explicitly known symmetries, we propose the spatio-temporal stabilizers method in Section 7.3. In particular, we present the analysis and the implementation of spatio-temporal stabilizers in Section 7.3.1 and 7.3.2, respectively, followed by our discussions of the associated practical issues, including the simultaneous observability and the accuracy vs. overhead trade-off in Section 7.3.3 and 7.3.4. We then discuss the strategies of applying the method of spatio-temporal stabilizers to practical quantum algorithms in Section 7.4. Our numerical results are discussed in Section 7.5, and finally, we conclude in Section 7.6.

## 7.1 State Symmetry and Circuit Symmetry

In this section, we discuss the difference between quantum state symmetries as well as circuit symmetries, and show the importance of distinguishing them from each other.

Let us consider the simple quantum circuit portrayed in Fig. 7.1. In this diagram, $\mathcal{R}_{\mathrm{x}}(\cdot)$ denotes a single-qubit X-rotation gate, while $\mathcal{R}_{\mathrm{xx}}(\cdot)$ denotes a two-qubit XX-rotation gate. Note that this circuit may be represented by an operator that is diagonal under the X-basis. To see this, recall that all Z-rotation gates are represented by diagonal matrices under the conventional computational basis, also known as the Z-basis. By the same token, all X-rotation gates are diagonal under the X-basis, since we could turn X-rotations into Z-rotations by changing the basis. When the input state of the circuit is

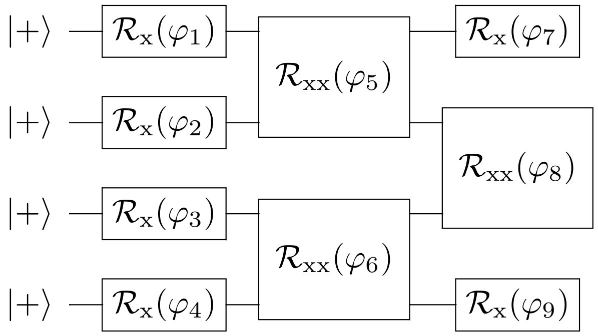


FIGURE 7.1: A quantum circuit having symmetries that may be viewed from both state-oriented and circuit-oriented perspectives.

$|+\rangle^{\otimes 4}$ as shown in the figure, we observe two different types of symmetries as follows:

- **State symmetry**: The output state of the circuit has the stabilizer $\mathcal{S} = \boldsymbol{X}_1\boldsymbol{X}_2\boldsymbol{X}_3\boldsymbol{X}_4$, where $\boldsymbol{X}_i$ denotes the Pauli-X operator acting on the $i$-th qubit. This stabilizer may be used to detect Z-errors.
- **Circuit symmetry**: This circuit can be diagonalized under the X-basis. Consequently, we have: 1) Every gate in this circuit commutes with one another; 2) The circuit commutes with the operator $\mathcal{S}$.

Observe that in this simple example, the circuit symmetries are more fundamental and more essential than the state symmetry. Indeed, the stabilizer $\mathcal{S}$ originates from the fact that the circuit commutes with $\mathcal{S}$, and that the input state is an eigenstate of $\mathcal{S}$. If the input state is different, the state may no longer be stabilized by $\mathcal{S}$, and hence symmetry verification techniques based on stabilizer checks are no longer applicable. However, the circuit symmetries are still valid in this case. This motivates us to design efficient techniques for verifying circuit symmetries and for mitigating errors that violate these symmetries.

## 7.2 Gate Commutativity Verification: Quantum Switch

In this section, we show that the commutativity of gates in a quantum circuit could by verified by exploiting the concept of quantum switches. Note that this is a weaker circuit symmetry compared to "the circuit commutes with some known operator", which will be investigated in the next section.

Quantum switches constitute a physical realization of the superposition of causal orders, producing quantum states that are coherent superpositions of the outputs of certain quantum circuits. These circuits contain the same operations, but are executed in different sequential orders. Quantum switches have received the attention of both communication and information theorists, since they have been shown to have the potential of improving the overall capacity by superposing certain noisy channels [200]. In its simplest form, the quantum switch involving a pair of channels $\mathcal{A}$ and $\mathcal{B}$ would effectively produce a superposition of $\mathcal{A} \circ \mathcal{B}$ and $\mathcal{B} \circ \mathcal{A}$, with the assistance of a control qubit. The composite channel may be represented as follows:

$$\mathcal{C}(\rho, \omega) = \sum_{i,j} \boldsymbol{C}_{ij}(\rho \otimes \omega)\boldsymbol{C}_{ij}^{\dagger}, \tag{7.1}$$

where $\rho$ and $\omega$ represent the state of the data register and the control qubit, respectively, while $\boldsymbol{C}_{ij}$ denotes a Kraus operator of $\mathcal{C}$ given by

$$\boldsymbol{C}_{ij} = \boldsymbol{A}_i\boldsymbol{B}_j \otimes |0\rangle\langle 0| + \boldsymbol{B}_j\boldsymbol{A}_i \otimes |1\rangle\langle 1|, \tag{7.2}$$

with $\boldsymbol{A}_i$ and $\boldsymbol{B}_j$ denoting the Kraus operators of $\mathcal{A}$ and $\mathcal{B}$, respectively. We observe from (7.2) that $\mathcal{A} \circ \mathcal{B}$ is applied when we measure a $|0\rangle$ on the control qubit, and $\mathcal{B} \circ \mathcal{A}$ is applied otherwise. This suggest that if the control qubit is set to be a superposition of $|0\rangle$ and $|1\rangle$, the resulting channel would be a superposition of $\mathcal{A} \circ \mathcal{B}$ and $\mathcal{B} \circ \mathcal{A}$. A classical example showing the information-theoretic advantage of the quantum switch is that, when both $\mathcal{A}$ and $\mathcal{B}$ are entanglement-breaking channels (which are extremely noisy) given by

$$\mathcal{A}(\rho) = \mathcal{B}(\rho) = \frac{1}{2}(\boldsymbol{X}\rho\boldsymbol{X} + \boldsymbol{Y}\rho\boldsymbol{Y}),$$

then we obtain a noiseless channel by performing post-selection based on the control qubit.

Inspired by the example of entanglement-breaking channels, we propose to verify the commutativity of gates using quantum switches. Intuitively, we first prepare the control qubit at a superposition state of $|0\rangle$ and $|1\rangle$ in order to produce a superposition of $\mathcal{A} \circ \mathcal{B}$ and $\mathcal{B} \circ \mathcal{A}$. Then, conditioned on the measured outcome of the control qubit, we discard the computational results corresponding to the non-commutative components. Formally speaking, we have the following result.

**Proposition 7.1** (Commutator-Omitting Switches). *Suppose that the control qubit is initialized to the state $|+\rangle$. If we do not discard any result, the state of the data register is*[1]

$$\rho_{\text{raw}} = \sum_{i,j} \frac{\{\boldsymbol{A}_i, \boldsymbol{B}_j\}}{2}\rho\frac{\{\boldsymbol{A}_i, \boldsymbol{B}_j\}^{\dagger}}{2} + \frac{[\boldsymbol{A}_i, \boldsymbol{B}_j]}{2}\rho\frac{[\boldsymbol{A}_i, \boldsymbol{B}_j]^{\dagger}}{2}. \tag{7.3}$$

---

[1] The commutator and the anti-commutator between two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are defined as $[\boldsymbol{A}, \boldsymbol{B}] := \boldsymbol{AB} - \boldsymbol{BA}$ and $\{\boldsymbol{A}, \boldsymbol{B}\} := \boldsymbol{AB} + \boldsymbol{BA}$, respectively.

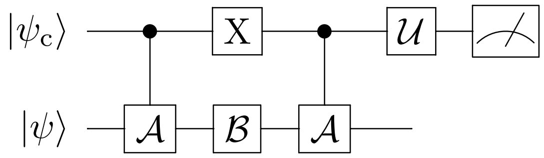


FIGURE 7.2: The quantum circuit implementation of a quantum switch between two commuting gates.

*By contrast, if we do discard the state once we measure a $|-\rangle$ at the output of the quantum switch, the state of the data register is given by*

$$\rho_{\text{out}} = \frac{1}{Z}\sum_{i,j}\frac{\{\boldsymbol{A}_i,\boldsymbol{B}_j\}}{2}\rho\frac{\{\boldsymbol{A}_i,\boldsymbol{B}_j\}^\dagger}{2}, \tag{7.4}$$

*where Z is a normalization factor given by*

$$Z = \frac{\text{Tr}\{\sum_{i,j}\{\boldsymbol{A}_i,\boldsymbol{B}_j\}\rho\{\boldsymbol{A}_i,\boldsymbol{B}_j\}^\dagger + [\boldsymbol{A}_i,\boldsymbol{B}_j]\rho[\boldsymbol{A}_i,\boldsymbol{B}_j]^\dagger\}}{\text{Tr}\{\sum_{i,j}\{\boldsymbol{A}_i,\boldsymbol{B}_j\}\rho\{\boldsymbol{A}_i,\boldsymbol{B}_j\}^\dagger\}}.$$

*Proof.* Please refer to Appendix A.12. □

From Proposition 7.1 we see that with the help of the quantum switch, we may filter out the components taking the form of $[\boldsymbol{A}_i,\boldsymbol{B}_j]\rho[\boldsymbol{A}_i,\boldsymbol{B}_j]^\dagger$ from the output state. Since $\mathcal{A}\circ\mathcal{B}$ should be equivalent to $\mathcal{B}\circ\mathcal{A}$ if both $\mathcal{A}$ and $\mathcal{B}$ are noiseless, we have $[\boldsymbol{A}_i,\boldsymbol{B}_j] = 0$ under the noise-free condition. This implies that by filtering out components like $[\boldsymbol{A}_i,\boldsymbol{B}_j]\rho[\boldsymbol{A}_i,\boldsymbol{B}_j]^\dagger$, we may mitigate the computational error. To elaborate further, let us consider the classical average of the computational results of $\mathcal{A}\circ\mathcal{B}$ and $\mathcal{B}\circ\mathcal{A}$, which may be expressed as

$$\rho_{\text{avg}} = \frac{1}{2}\sum_{i,j}\boldsymbol{A}_i\boldsymbol{B}_j\rho\boldsymbol{B}_j^\dagger\boldsymbol{A}_i^\dagger + \boldsymbol{B}_j\boldsymbol{A}_i\rho\boldsymbol{A}_i^\dagger\boldsymbol{B}_j^\dagger. \tag{7.5}$$

After some further manipulations, one would obtain $\rho_{\text{avg}} = \rho_{\text{raw}}$. This means that by combining a quantum switch and post-selection, we could indeed eliminate certain error components in the raw output state that do not satisfy the gate commutativity conditions.

### 7.2.1 Circuit Implementation and Practical Issues

The quantum switch between two commuting gates $\mathcal{A}$ and $\mathcal{B}$ can be implemented with the aid of a control qubit [77], as portrayed in Fig. 7.2. The states $|\psi_c\rangle$ and $|\psi\rangle$ represent the states of the control qubit and that of the data register, respectively. The gate $\mathcal{U}$ is applied for rotating the control qubit so that its state becomes diagonal under the

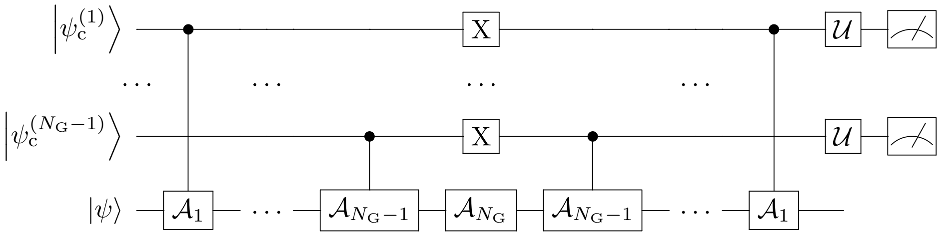


FIGURE 7.3: The implementation of the quantum switch methods containing $N_G > 2$ commuting gates relying on multiple control qubits.

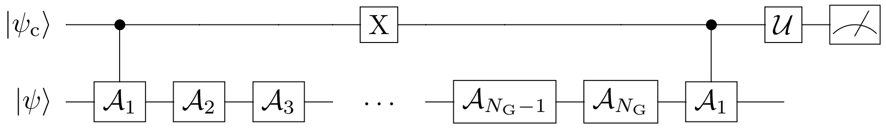


FIGURE 7.4: The implementation of the quantum switch methods containing $N_G > 2$ commuting gates using a single control qubit.

Z-basis. For example, the control qubit is typically initialized to the state $|\psi_c\rangle = |+\rangle$, and thus the corresponding $\mathcal{U}$ is the Hadamard gate. Upon measuring a $|0\rangle$ on the control qubit, we know that the commutativity between gates $\mathcal{A}$ and $\mathcal{B}$ is preserved. Otherwise, we discard the computational result.

There are some noteworthy issues associated with this implementation, when we apply it to practical quantum circuits. First of all, one of the two gates (e.g. the gate $\mathcal{A}$ in Fig. 7.2) has to be implemented in a controlled form, which increases the number of qubits that it acts upon. In practice, a quantum gate acting on more qubits is typically noisier than those acting on less qubits. Therefore, it is not clear whether the quantum switch method achieves a practical accuracy improvement over the original (unprotected) circuit. Another issue is that there is no natural and unified generalization of the method to $N_G > 2$ gates under the gate model.[2] Here we present some possible generalizations relying on multiple control qubits, portrayed in Fig. 7.3 and 7.4.

## 7.3 Commutativity with Known Unitaries: Spatio-Temporal Stabilizers

In the previous section, we have shown that quantum switches could be used to verify the commutativity of quantum gates. But in some practical scenarios, we may have a stronger circuit symmetry, in the sense that a block of gates commute with some known unitaries. For example, in the QAOA, the part implementing a phase Hamiltonian commutes with all Pauli operators containing only Pauli-I and Pauli-Z operators.

---

[2] Natural generalizations do exist for other models of quantum computation, for example, photonic quantum computers using the implementation described in [78].

Intuitively, this stronger sense of symmetry may lead to better error mitigation performance than that of gate commutativity.

### 7.3.1 Improving the Quantum Switch Method

In fact, we could verify this strong sense of circuit symmetry by slightly modifying the quantum switch method. Let us denote the circuit to be verified as $\mathcal{C}(\rho) = \sum_i \boldsymbol{C}_i \rho \boldsymbol{C}_i^\dagger$, and assume that the noiseless component in the circuit, represented by the Kraus operator $\boldsymbol{C}_1$, commutes with the operator $\mathcal{U}(\rho) = \boldsymbol{U}\rho\boldsymbol{U}^\dagger$. By applying a quantum switch between $\mathcal{C} \circ \mathcal{U}$ and $\mathcal{U} \circ \mathcal{C}$, we obtain the following composite circuit

$$\mathcal{D}(\rho, \omega) = \sum_i \boldsymbol{D}_i (\rho \otimes \omega) \boldsymbol{D}_i^\dagger, \tag{7.6}$$

where

$$\boldsymbol{D}_i = \boldsymbol{C}_i \boldsymbol{U} \otimes |0\rangle\langle 0| + \boldsymbol{U}\boldsymbol{C}_i \otimes |1\rangle\langle 1| \,. \tag{7.7}$$

Similar to the result in Proposition 7.1, after applying $\mathcal{D}$, the output state is given by

$$\rho_{\mathrm{m}} \propto \sum_i \frac{\{\boldsymbol{C}_i, \boldsymbol{U}\}}{2} \rho \frac{\{\boldsymbol{C}_i, \boldsymbol{U}\}^\dagger}{2}. \tag{7.8}$$

Now we have a coherent superposition of $\mathcal{C} \circ \mathcal{U}$ and $\mathcal{U} \circ \mathcal{C}$. But in order to verify the strong circuit symmetry, we do not need to actually apply $\mathcal{U}$, which differs from the case discussed in the previous section. In light of this, we apply the inverse of $\mathcal{U}$, namely $\mathcal{U}^\dagger$, to $\rho_{\mathrm{m}}$ and obtain the final output as

$$\rho_{\mathrm{out}} \propto \sum_i \frac{\boldsymbol{C}_i + \boldsymbol{U}^\dagger \boldsymbol{C}_i \boldsymbol{U}}{2} \rho \frac{\boldsymbol{C}_i^\dagger + \boldsymbol{U}^\dagger \boldsymbol{C}_i^\dagger \boldsymbol{U}}{2}. \tag{7.9}$$

In this way, we eliminate the impact of $\mathcal{U}$ on the noiseless component in the final result by exploiting the commutativity between $\boldsymbol{U}$ and $\boldsymbol{C}_1$. Indeed, observe from (7.9) that for the noiseless component $\boldsymbol{C}_1$, we have

$$\frac{\boldsymbol{C}_1 + \boldsymbol{U}^\dagger \boldsymbol{C}_1 \boldsymbol{U}}{2} = \boldsymbol{C}_1,$$

since $\boldsymbol{U}\boldsymbol{C}_1 = \boldsymbol{C}_1\boldsymbol{U}$, implying that it remains unchanged by our modified quantum switch.

We could gain further insights into the error mitigation performance of this modified quantum switch by considering more specific noise models. Here we assume that each Kraus operator $\boldsymbol{C}_i$ can be decomposed as $\widetilde{\boldsymbol{C}}_i \boldsymbol{C}_1$, namely the noiseless circuit followed by some quantum channel modeling the noise. For the noiseless component we have

$\widetilde{C}_1 = \boldsymbol{I}$. Under these assumptions, we may obtain

$$\begin{aligned} C_i + \boldsymbol{U}^\dagger C_i \boldsymbol{U} &= \widetilde{C}_i C_1 + \boldsymbol{U}^\dagger \widetilde{C}_i C_1 \boldsymbol{U} \\ &= \left(\widetilde{C}_i + \boldsymbol{U}^\dagger \widetilde{C}_i \boldsymbol{U}\right) C_1. \end{aligned} \tag{7.10}$$

Let us further assume that the symmetry operator $\boldsymbol{U}$ is a Pauli operator, which is common for practical quantum circuits. Note that among the group of Pauli operators, given a fixed operator $\boldsymbol{U}$, any other operator either commutes with $\boldsymbol{U}$ or anti-commutes with $\boldsymbol{U}$. This implies that $\widetilde{C}_i$ may be decomposed into two parts as

$$\widetilde{C}_i = \widetilde{C}_i^{(\mathrm{comm})} + \widetilde{C}_i^{(\mathrm{anti})}, \tag{7.11}$$

where $\widetilde{C}_i^{(\mathrm{comm})}$ commutes with $\boldsymbol{U}$ and $\widetilde{C}_i^{(\mathrm{anti})}$ anti-commutes with $\boldsymbol{U}$. This is because all quantum operations can be represented as linear combinations of Pauli operators. Therefore, (7.10) can be further simplified as

$$\begin{aligned} C_i + \boldsymbol{U}^\dagger C_i \boldsymbol{U} = &\left(\widetilde{C}_i^{(\mathrm{comm})} + \widetilde{C}_i^{(\mathrm{anti})}\right) C_1 \\ &+ \left(\widetilde{C}_i^{(\mathrm{comm})} - \widetilde{C}_i^{(\mathrm{anti})}\right) C_1 \\ = &\, 2\widetilde{C}_i^{(\mathrm{comm})} C_1, \end{aligned} \tag{7.12}$$

since

$$\boldsymbol{U}^\dagger \widetilde{C}_i^{(\mathrm{anti})} \boldsymbol{U} = -\widetilde{C}_i^{(\mathrm{anti})},$$

and

$$\boldsymbol{U}^\dagger \widetilde{C}_i^{(\mathrm{comm})} \boldsymbol{U} = \widetilde{C}_i^{(\mathrm{comm})}.$$

Hence we have

$$\rho_{\mathrm{out}} \propto \sum_i \widetilde{C}_i^{(\mathrm{comm})} C_1 \rho C_1^\dagger \left(\widetilde{C}_i^{(\mathrm{comm})}\right)^\dagger. \tag{7.13}$$

One could verify that similar arguments can also be applied to the case where $\mathcal{C}$ consists of more than one noisy gates. For example, when there are two noisy gates in the circuit, the Kraus operators satisfy $C_{ij} = \widetilde{C}_{1,i} C_{1,1} \widetilde{C}_{2,j} C_{2,1}$, and we have

$$\begin{aligned} &C_{ij} + \boldsymbol{U}^\dagger C_{ij} \boldsymbol{U} \\ &= \widetilde{C}_{1,i}^{(\mathrm{comm})} C_{1,1} \widetilde{C}_{2,i}^{(\mathrm{comm})} C_{2,1} + \widetilde{C}_{1,i}^{(\mathrm{anti})} C_{1,1} \widetilde{C}_{2,i}^{(\mathrm{anti})} C_{2,1}, \end{aligned} \tag{7.14}$$

as long as both $C_{1,1}$ and $C_{2,1}$ commute with $\boldsymbol{U}$. If we assume that the channels only impose anti-commutative errors (e.g., bit-flip channels when $\boldsymbol{U} = \boldsymbol{Z}$), and that the anti-commutative Kraus operators such as $\widetilde{C}_{1,i}^{(\mathrm{anti})}$ (and also others with different subscripts) satisfy $\|\widetilde{C}_{1,i}^{(\mathrm{anti})}\| = O(\sqrt{\epsilon})$ where $\epsilon$ denotes the average error rate per gate, we may infer from (7.14) that:

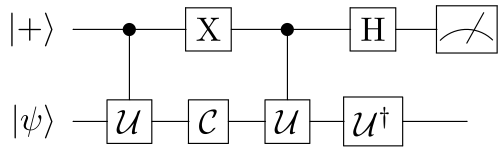


FIGURE 7.5: Direct implementation of an STS check.

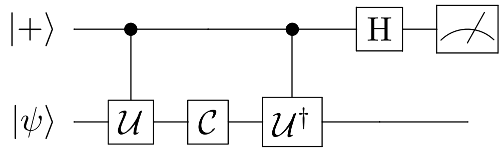


FIGURE 7.6: Simplified implementation of an STS check.

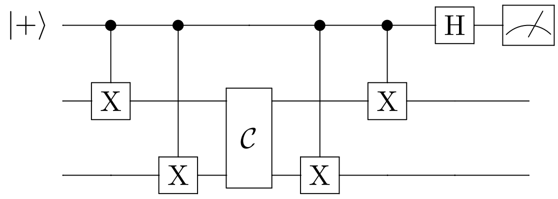


FIGURE 7.7: The STS $\mathcal{S}\{\boldsymbol{X}_1(0), \boldsymbol{X}_2(0), \boldsymbol{X}_1(1), \boldsymbol{X}_2(1)\}$.

*Remark* 7.2. Upon the verification of the commutativity with $\mathcal{U}$, the residual error for a circuit containing multiple noisy gates is on the order of $O(\epsilon^2)$.

Intuitively, by verifying a circuit symmetry $\mathcal{U}$ which is a Pauli operator, we may detect any single-qubit anti-commutative error. This resembles the characteristics of error-detecting stabilizer codes. Partly for this reason, we will refer to the aforementioned modified quantum switch method as the "spatio-temporal stabilizer method" in the rest of this chapter. This terminology will be explained in more detail in Section 7.3.2.

### 7.3.2 Implementation: Spatio-temporal Stabilizer Check

According to the discussion in Section 7.3.1, we could readily obtain a circuit implementing the modified quantum switch portrayed in Fig. 7.5. But this circuit admits a simplification, as portrayed in Fig. 7.6, which helps us better understand this method. As it may be observed from the figure, the final state of the data register would be $\mathcal{U}^\dagger \circ \mathcal{C} \circ \mathcal{U}(|\psi\rangle\langle\psi|)$ if the control qubit is in $|1\rangle$, when the controlled-$\mathcal{U}$ and controlled-$\mathcal{U}^\dagger$ gates are being applied, and $\mathcal{C}(|\psi\rangle\langle\psi|)$ if the control qubit is in $|0\rangle$. But the control qubit is in $|+\rangle$ due to the Hadamard gate, hence if we measure a $|0\rangle$ on the control qubit at the output of the circuit, the Kraus operators on the data register are given by

$$\boldsymbol{K}_i = \frac{1}{2}\left(\boldsymbol{C}_i + \boldsymbol{U}^\dagger \boldsymbol{C}_i \boldsymbol{U}\right), \tag{7.15}$$

as we have expected.

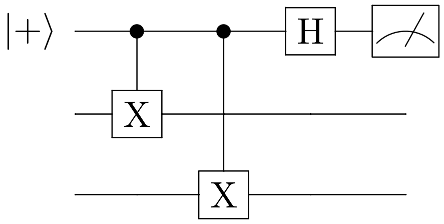


FIGURE 7.8: The conventional stabilizer $X_1X_2$.

To gain further intuition, we consider a toy example, where the circuit $\mathcal{C}$ commutes with a Pauli operator $\mathcal{U}$ given by $\boldsymbol{U} = \boldsymbol{X}_1\boldsymbol{X}_2$. In this case, the simplified circuit can be constructed as shown in Fig. 7.7. From this figure we see that the simplified circuit is rather similar to the ones performing stabilizer checks. For example, if we wish to measure a stabilizer $\boldsymbol{X}_1\boldsymbol{X}_2$, we could use the circuit portrayed in Fig. 7.8. Compared to Fig. 7.8, the circuit in Fig. 7.7 looks like measuring a bipartite "stabilizer", for which a part is applied before the circuit $\mathcal{C}$, and the rest of it is applied after $\mathcal{C}$. In fact, upon denoting the input state of the data register as $|\psi\rangle$, it is clear that the output state $\boldsymbol{C}|\psi\rangle$ has the following stabilizer

$$\boldsymbol{S} = \boldsymbol{C}(\boldsymbol{X}_1\boldsymbol{X}_2)\boldsymbol{C}^\dagger(\boldsymbol{X}_1\boldsymbol{X}_2), \tag{7.16}$$

and that the circuit in Fig. 7.7 indeed measures the stabilizer $\boldsymbol{S}$. Since the gates in quantum circuits are executed in a sequential manner, if we define the time right before $\mathcal{C}$ is applied as $t = 0$, and the time right after $\mathcal{C}$ is applied as $t = 1$, we see that the stabilizer $\boldsymbol{S}$ contains a $(\boldsymbol{X}_1\boldsymbol{X}_2)$ at time $t = 0$, and another $(\boldsymbol{X}_1\boldsymbol{X}_2)$ at time $t = 1$. Therefore, we refer to $\boldsymbol{S}$ as a *"spatio-temporal stabilizer"* of the output state $\boldsymbol{C}|\psi\rangle$, which can be formally defined as follows.

**Definition 7.3** (Spatio-temporal Stabilizer)**.** Consider a quantum circuit consisting of $N$ unitary gates given by $\boldsymbol{C} = \boldsymbol{C}_N\boldsymbol{C}_{N-1}\ldots\boldsymbol{C}_1$, with input state $|\psi\rangle$. We say that $\boldsymbol{S}$ is a $(N+1)$-partite spatio-temporal stabilizer (STS) of the output state $\boldsymbol{C}|\psi\rangle$, if it satisfies $\boldsymbol{S}\boldsymbol{C}|\psi\rangle = \boldsymbol{C}|\psi\rangle$, and takes the following form

$$\begin{aligned} \boldsymbol{S} &= \mathcal{S}\{\boldsymbol{S}_0(0), \boldsymbol{S}_1(1), \ldots, \boldsymbol{S}_N(N)\} \\ &:= \boldsymbol{C}\boldsymbol{S}_0^\dagger\boldsymbol{C}_1^\dagger\boldsymbol{S}_1^\dagger\ldots\boldsymbol{C}_N^\dagger\boldsymbol{S}_N^\dagger. \end{aligned} \tag{7.17}$$

The argument $t$ in $\boldsymbol{S}_n(t)$ represents the time instance when this partial operator is applied. The partial operators $\boldsymbol{S}_n$, $n = 1\ldots N$ are called the *components* of $\boldsymbol{S}$. When it is more convenient, we may require that the partial operators $\{\boldsymbol{S}_n\}_{n=0}^N$ satisfy the following (stronger) alternative condition

$$\boldsymbol{S}_N\boldsymbol{C}_N\boldsymbol{S}_{N-1}\ldots\boldsymbol{C}_1\boldsymbol{S}_0 = \boldsymbol{C}. \tag{7.18}$$

We say that $\boldsymbol{C}$ is an STS *of the circuit* if (7.18) is satisfied. In the context of STSs, we refer to the control qubits as *ancillas* to be consistent with the terminologies in the conventional

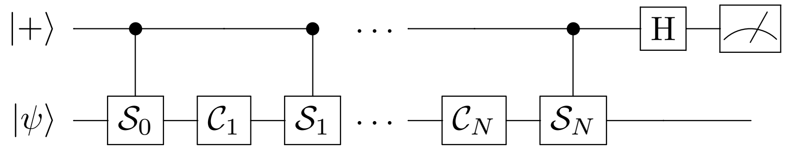


FIGURE 7.9: The circuit measuring the STS in (7.17).

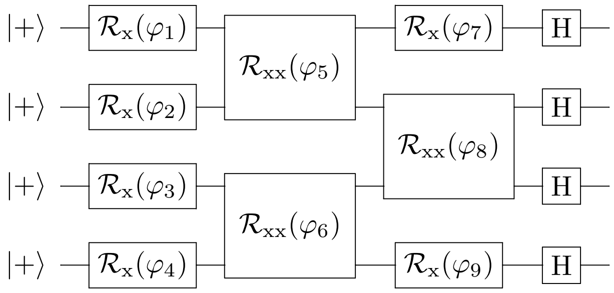


FIGURE 7.10: A circuit having an STS as in (7.19), but it is difficult to find an operator that commutes with it.

stabilizer formalism.

The circuit measuring the STS in (7.17) is portrayed in Fig. 7.9. We may observe from Fig. 7.9 that the concept of STS actually generalizes the idea of verifying circuit commutativity with known operators, since the partial operators $\mathcal{S}_0$ through $\mathcal{S}_N$ can all be different. A natural question that arises is, whether this generalization has any practical implication. In fact, we could illustrate the usefulness of this generalization, by revisiting the example in Fig. 7.1. We now see that the circuit commutes with $\boldsymbol{X}^{\otimes 4}$, and equivalently, we say that the circuit has the STS

$$\begin{aligned}\mathcal{S}^{(\mathrm{st})}\{&\boldsymbol{X}_1(0),\boldsymbol{X}_2(0),\boldsymbol{X}_3(0),\boldsymbol{X}_4(0),\\&\boldsymbol{X}_1(1),\boldsymbol{X}_2(1),\boldsymbol{X}_3(1),\boldsymbol{X}_4(1)\}.\end{aligned}$$

But if we further apply a Hadamard gate to each of the qubits at the output of the circuit, as portrayed in Fig. 7.10, it becomes difficult to find an operator that commutes with the new circuit. By contrast, we could say that this circuit has a different STS given by

$$\begin{aligned}\boldsymbol{S}=\mathcal{S}^{(\mathrm{st})}\{&\boldsymbol{X}_1(0),\boldsymbol{X}_2(0),\boldsymbol{X}_3(0),\boldsymbol{X}_4(0),\\&\boldsymbol{Z}_1(1),\boldsymbol{Z}_2(1),\boldsymbol{Z}_3(1),\boldsymbol{Z}_4(1)\},\end{aligned}\tag{7.19}$$

since the circuit (denoted by $\boldsymbol{C}$) satisfies

$$\boldsymbol{Z}^{\otimes 4}\boldsymbol{C}=\boldsymbol{C}\boldsymbol{X}^{\otimes 4}.\tag{7.20}$$

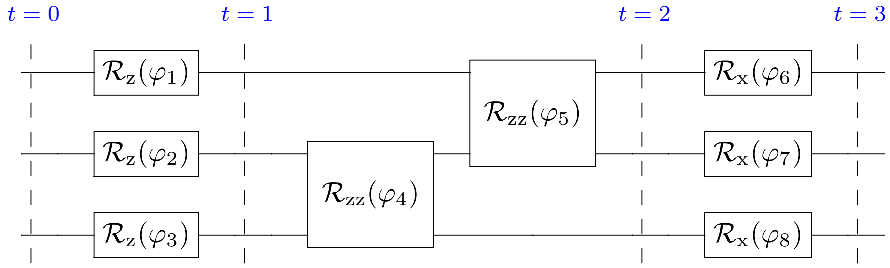


FIGURE 7.11: A circuit having two STSs $S_1 = \mathcal{S}\{Z^{\otimes 3}(0), Z^{\otimes 3}(2)\}$ and $S_2 = \mathcal{S}\{X^{\otimes 3}(1), X^{\otimes 3}(3)\}$ that are not simultaneously observable.

### 7.3.3 Simultaneous Observability of STSs

When we consider the verification of a quantum state or a circuit that has multiple symmetries, a natural requirement is that these symmetries can be checked at the same time. Otherwise, only a subset of the symmetries can be verified in each computation, which may result in an unsatisfactory error mitigation performance.

Simultaneous observability is a natural property of conventional stabilizers [48, Sec. 10.5.4]. A fundamental characteristic of quantum mechanics is the uncertainty principle, stating that a pair of observables can be simultaneously determined to an arbitrary accuracy, if and only if they commute with each other. Stabilizers, being special cases of observables, also follow this principle. In fact, all stabilizers of the same quantum state commute with one another, and hence they form the so-called stabilizer group [48, Sec. 10.5.4]. This is easily seen by observing that

$$\begin{aligned} &S_1 |\psi\rangle = |\psi\rangle \text{ AND } S_2 |\psi\rangle = |\psi\rangle \\ &\quad \Longrightarrow S_1 S_2 |\psi\rangle = S_2 S_1 |\psi\rangle = |\psi\rangle, \\ &S_1 S_2 \neq S_2 S_1 \Longrightarrow S_1 S_2 |\psi\rangle \neq S_2 S_1 |\psi\rangle. \end{aligned}$$

Therefore, conventional stabilizers of the same state are always simultaneously observable.

For STSs, however, simultaneous observability is not necessarily satisfied. To be more specific, let us consider the example portrayed in Fig. 7.11. It is clear that the circuit has two STSs, namely $S_1 = \mathcal{S}\{Z^{\otimes 3}(0), Z^{\otimes 3}(2)\}$ and $S_2 = \mathcal{S}\{X^{\otimes 3}(1), X^{\otimes 3}(3)\}$. However, $S_1$ and $S_2$ are not simultaneously observable, since $X^{\otimes 3}$ does not commute with $Z^{\otimes 3}$, and hence the combination of $S_1$ and $S_2$ given by $\mathcal{S}\{Z^{\otimes 3}(0), X^{\otimes 3}(1), Z^{\otimes 3}(2), X^{\otimes 3}(3)\}$ is not an STS of the original circuit. Therefore, we are motivated to propose the following formal definition of simultaneous observability for STSs.

**Definition 7.4** (Simultaneous Observability)**.** Consider a set of STS checks of a certain circuit $\mathcal{C}$, implemented in the fashion shown in Fig. 7.9 with the aid of ancillas. If the state of the data register at the output of $\mathcal{C}$ is the same regardless of the initial states of the ancillas, we say that the STSs are simultaneously observable.

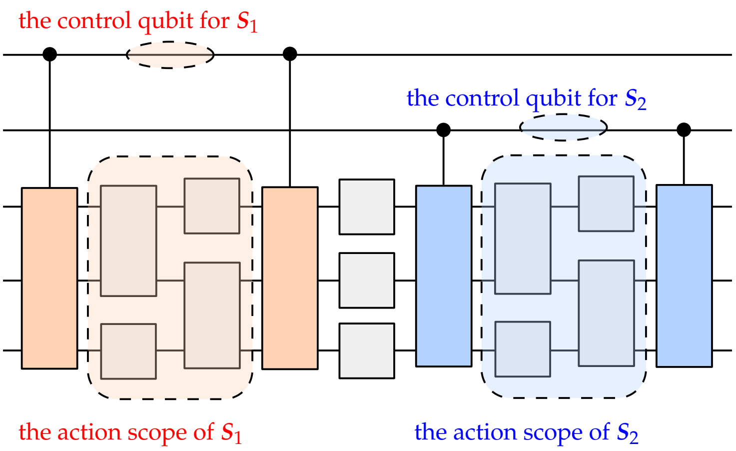


FIGURE 7.12: STSs having temporally disjoint action scopes are simultaneously observable.

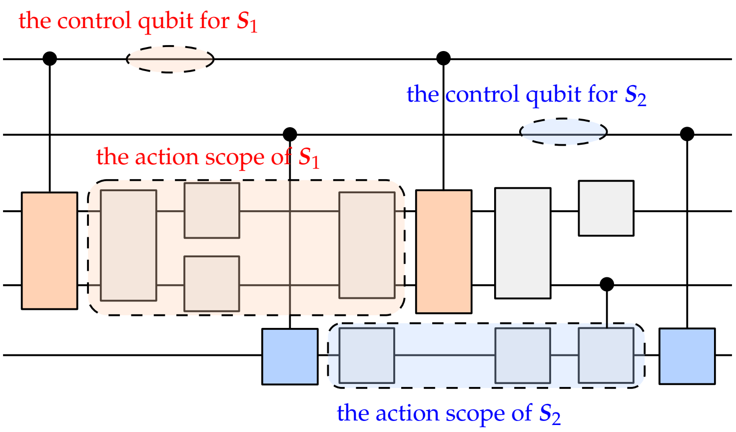


FIGURE 7.13: STSs having spatially disjoint action scopes are simultaneously observable.

Intuitively, by initializing some ancillas to the state $|0\rangle$, we effectively "disable" certain STSs. Hence, simultaneous observability means that an arbitrary combination of the STSs still constitutes an STS of the circuit. Unfortunately, determining the simultaneous observability directly using the definition may be inconvenient when the number of STSs is large, given the excessive number of possible STSs combinations. To this end, we provide some sufficient conditions that may be useful in practice, based on the following definition of the *action scope* of STSs.

**Definition 7.5** (Action Scopes)**.** The action scope of an STS $S$ is a set $\mathscr{S} = \mathscr{S}_s \times \mathscr{S}_t$, where $\mathscr{S}_s$ is the spatial action scope constituted by the indices of all qubits that the component operators of $S$ act upon, while $\mathscr{S}_t = \{t | t \leq t_{\max}, \ t \geq t_{\min}, \ t \in \mathbb{Z}\}$ is the temporal action scope, with $t_{\max}$ and $t_{\min}$ denoting the maximum and the minimum temporal indices in $S$, respectively.

To elaborate, for example, the action scope of the STS $S = \mathcal{S}\{X_0(0), Z_1(0), X_1(2), Z_2(3)\}$

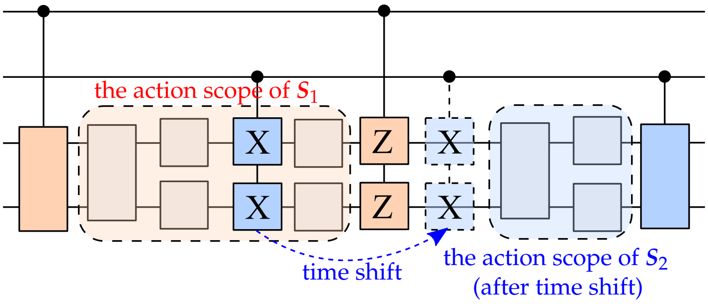


FIGURE 7.14: STSs having disjoint action scopes after appropriate time shifts are simultaneously observable.

is $\{0,1,2\} \times \{0,1,2,3\}$. By exploiting the concept of action scope, the following sufficient condition of simultaneous observability may be obtained.

*Sufficient Condition* 1 (Disjoint Action Scopes). If the action scopes of a set of STSs are mutually disjoint, these STSs are simultaneously observable.

*Proof.* If the STSs $S_1$ and $S_2$ have disjoint action scopes, they can be viewed as STSs of two disjoint sub-circuits of the original circuit, respectively, as portrayed in Fig. 7.12. Hence they are simultaneously observable. □

A more sophisticated (and potentially more useful) sufficient condition may be obtained by modifying Sufficient Condition 1, detailed as follows.

*Sufficient Condition* 2 (Disjoint Action Scopes After Time Shift). Consider a set of STSs $\mathscr{A}$. The STSs in $\mathscr{A}$ are simultaneously observable, if for each $S_i \in \mathscr{A}$, we may impose appropriate "time shifts" to $\forall S_j \in \mathscr{A}$, $j \neq i$, ensuring that the results after the time shifts are still STSs of the original circuit, and that their action scopes are disjoint with that of $S_i$. A legitimate time shift for STS $S_j$ is a translation of certain components in $S_j$ to another time instance, satisfying the condition that these components commute with all the components of other STSs in $\mathscr{A}$ lying on the trajectory of the translation, as portrayed in Fig. 7.14.

*Proof.* Denote the result of time shift for $S_j$ as $\mathcal{T}(S_j)$. From Sufficient Condition 1 we see that $\mathcal{T}(S_j)$ and $S_i$ are simultaneously observable, and thus the combination of $S_i$ and $\mathcal{T}(S_j)$ is an STS. Since the translated components of $S_j$ commute with those of other STSs on the translation trajectory, we see that the combination of $S_i$ and $S_j$ is also an STS. By applying the arguments to all pairs of STSs in $\mathscr{A}$, we arrive at the desired result. □

In the example shown in Fig. 7.14, the STSs $S_1$ and $S_2$ are simultaneously observable, because $X^{\otimes 2}$ commutes with $Z^{\otimes 2}$. We will see how this is related to the STSs of the QAOA in Section 7.4.3.

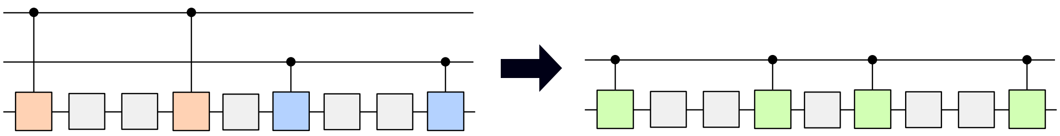


FIGURE 7.15: Reducing the overhead of control qubits by combining simultaneously observable STSs.

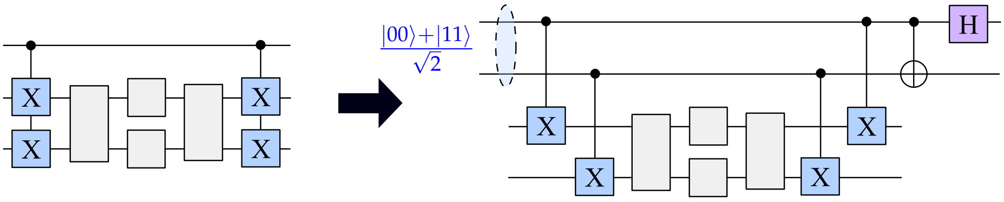


FIGURE 7.16: Mitigating error proliferation by measuring a single STS relying on multiple control qubits.

### 7.3.4 The Accuracy vs. Overhead Trade-off

According to the discussion in Section 7.3.2, by default, we use one ancilla for checking each STS. In fact, we could reallocate the qubit resources exploited for controlling STSs to strike more flexible accuracy vs. overhead trade-offs. For example, we may combine several simultaneously observable STSs into a single STS to reduce the overall qubit overhead, as portrayed in Fig. 7.15.

The overhead reduction obtained by combining STSs comes at a price of stronger error proliferation. To elaborate, observe that in the circuits shown in Fig. 7.15, the errors may propagate from the ancillas to the data register. However, the circuit on the right hand side suffers from more severe error proliferation, since the errors in the data register may propagate to the control, and then back to the data register. Therefore, when a higher accuracy is required and the qubit resources are abundant, we may measure a single STS using multiple ancillas to mitigate error proliferation, relying on pre-shared entanglements between the ancillas (i.e., the "cat" state [48, Sec. 10.6.3]), as portrayed in Fig. 7.16. This implementation bears some similarity with the fault-tolerant measurements of conventional stabilizers [48, Sec. 10.6.3].

Another type of computational overhead is the sampling overhead, which originates from the fact that some computational results are discarded due to their failure to pass the STS checks. To quantify the sampling overhead, we introduce the concept of sampling overhead factor, originally defined in [1] for the analysis of channel inversion-based QEM.

**Definition 7.6** (Sampling Overhead Factor)**.** The sampling overhead factor of a set $\mathscr{A}$ of STSs applied to a circuit $\mathcal{C}$ is defined as

$$\mathrm{SOF}(\mathcal{C},\mathscr{A})=\frac{1}{p_{\mathrm{pass}}(\mathcal{C},\mathscr{A})}-1, \tag{7.21}$$

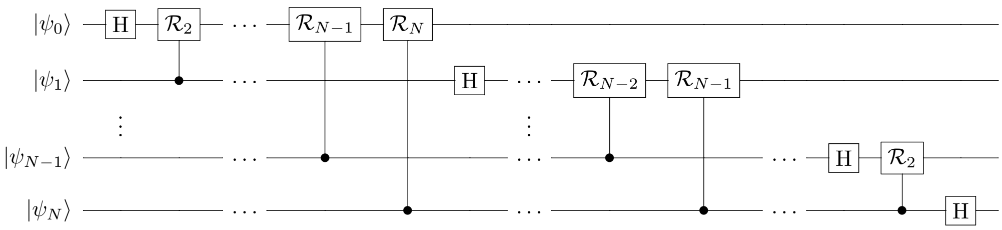


FIGURE 7.17: The circuit implementing an $N$-qubit QFT.

where $p_{\text{pass}}(\mathcal{C}, \mathscr{A})$ denotes the probability that the circuit passes all the STS checks in $\mathscr{A}$.

We will characterize the sampling overhead factors of the STSs applied to some practical quantum circuits in Section 7.5.

## 7.4 Case Study: The STSs of the QFT and the QAOA

In this section, we demonstrate the applicability and the characteristics of the STS method using two classes of practical quantum circuits, namely that of the QFT and the QAOA.

### 7.4.1 The STSs of the QFT Circuits

The QFT serves as a subroutine in the quantum phase estimation algorithm, which in turn plays significant roles in other more sophisticated quantum algorithms, including Shor's algorithm and the Harrow-Hassidim-Lloyd (HHL) algorithm [12,16]. Therefore, mitigating the error in the QFT is beneficial for a range of quantum algorithms.

The structure of an $N$-qubit QFT circuit is portrayed in Fig. 7.17, where the operator $\mathcal{R}_n$ (in the controlled-$\mathcal{R}_n$ gates) is a single-qubit Z-rotation defined by $\mathcal{R}_n = |0\rangle\langle 0| + e^{\imath 2\pi 2^{-n}} |1\rangle\langle 1|$. It is clearly seen from the figure that each qubit in the circuit participates in $(N-1)$ two-qubit controlled gates. For the gates before the Hadamard gate, the qubit serves as the control, while for those after the Hadamard gate, the qubit serves as the target.

We observe that for each qubit, the gates before the Hadamard gate and those after the Hadamard gate commute with the Pauli-Z operator, respectively, because all the two-qubit gates are controlled Z-rotations. Hence a straightforward implementation of the STSs is to treat these two blocks of gates separately, as shown in Fig. 7.18. However, this implementation may be excessively complex, since we would need two ancillas for every data qubit. Thus we may combine both STSs on each qubit, and arrive at the

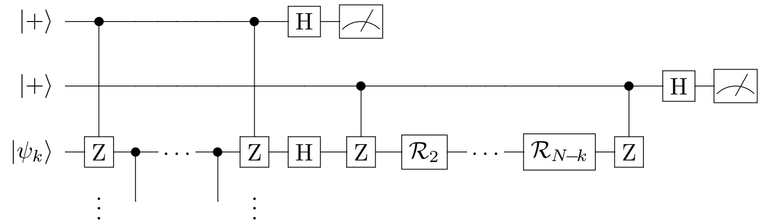


FIGURE 7.18: The straightforward implementation of the STSs on the single qubit in the QFT circuit.

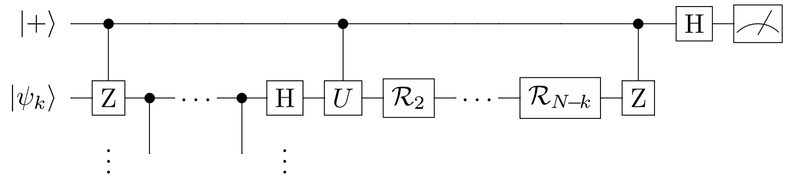


FIGURE 7.19: Combinging the STSs on the single qubit in the QFT circuit.

design portrayed in Fig. 7.19 after a slight simplification. The operator $\mathcal{U}$ in Fig. 7.19 has the following matrix representation

$$\boldsymbol{U} = \boldsymbol{Z}\boldsymbol{X} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \tag{7.22}$$

which only differs from the Pauli-Y operator by a global phase. Note that this global phase is non-negligible in the controlled-$\mathcal{U}$ operation.

### 7.4.2 Brief Introduction to the QAOA

The QAOA is an algorithm aiming for approximately solving discrete optimization problems taking the following form

$$\begin{aligned} &\max_{\boldsymbol{x}} \; F(\boldsymbol{x}) := \sum_{k=1}^{K} f_k(\boldsymbol{x}), \\ &\text{subject to } \; x_i \in \{-1, 1\}, \; \forall i = 1 \ldots N, \end{aligned} \tag{7.23}$$

where $\boldsymbol{x} = [x_1 \; \ldots \; x_N]^{\mathrm{T}}$, and $f_k(\boldsymbol{x})$ is a $k$-th order polynomial containing only $k$-th order monomials. For example, when $N = 3$, we may have $f_1(\boldsymbol{x}) = 0.1x_1 + 0.2x_2 + 0.3x_3$, $f_2(\boldsymbol{x}) = 0.4x_1x_2 + 0.5x_2x_3$, and $f_3(\boldsymbol{x}) = x_1x_2x_3$. The most common problem instances belong to the class of quadratic unconstrained binary optimization (QUBO) problems corresponding to $K = 2$, which can be expressed as

$$\begin{aligned} &\max_{\boldsymbol{x}} \; \boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{x}, \\ &\text{subject to } \; x_i \in \{-1, 1\}, \; \forall i = 1 \ldots N. \end{aligned} \tag{7.24}$$

By representing the vector $\boldsymbol{x}$ using a quantum state $|\psi\rangle$, we could represent the objective function $F(\boldsymbol{x})$ of (7.23) in the following alternative form

$$F(|\psi\rangle) = \langle\psi| \boldsymbol{H}_{\mathrm{P}} |\psi\rangle , \tag{7.25}$$

where $\boldsymbol{H}_{\mathrm{P}} = \sum_{n=1}^{K} \boldsymbol{F}_k$ is called the phase Hamiltonian encoding of the objective function, and $\boldsymbol{F}_k$ is the operator obtained by replacing terms such as $x_i$ in $f_k(\boldsymbol{x})$ by Pauli-Z operators $\boldsymbol{Z}_i$.

In order to maximize the objective function $F(|\psi\rangle)$, the QAOA applies two Hamiltonians, namely the phase Hamiltonian and the mixing Hamiltonian, in an alternating order. Specifically, given the initial state $|\psi(0)\rangle$, the output state can be expressed as

$$|\psi(\boldsymbol{\beta},\boldsymbol{\gamma})\rangle = e^{-\imath\beta_p \boldsymbol{H}_{\mathrm{M}}} e^{-\imath\gamma_p \boldsymbol{H}_{\mathrm{P}}} \ldots e^{-\imath\beta_1 \boldsymbol{H}_{\mathrm{M}}} e^{-\imath\gamma_1 \boldsymbol{H}_{\mathrm{P}}} |\psi(0)\rangle , \tag{7.26}$$

where $\boldsymbol{\beta} = [\beta_1 \ \ldots \ \beta_p]^{\mathrm{T}}$ and $\boldsymbol{\gamma} = [\gamma_1 \ \ldots \ \gamma_p]^{\mathrm{T}}$ are adjustable parameters controlling the search trajectory of the algorithm, and the mixing Hamiltonian $\boldsymbol{H}_{\mathrm{H}}$ is given by

$$\boldsymbol{H}_{\mathrm{M}} = \sum_{i=1}^{N} \boldsymbol{X}_i. \tag{7.27}$$

It has been shown that the optimal solution can be closely approximated by measuring $|\psi(\boldsymbol{\beta},\boldsymbol{\gamma})\rangle$ on the computational basis, when $p$ is sufficiently large and the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are chosen appropriately [32].

### 7.4.3 The STSs of the QAOA Circuits

From (7.26) we could observe that a typical QAOA circuit has $p$ stages, among which the $n$-th stage is

$$\boldsymbol{U}_n(\beta_n,\gamma_n) = e^{-\imath\beta_n \boldsymbol{H}_{\mathrm{M}}} e^{-\imath\gamma_n \boldsymbol{H}_{\mathrm{P}}}. \tag{7.28}$$

Since the structure of each stage is similar, we will focus on a single stage in the following analysis. It is clear that $\boldsymbol{H}_{\mathrm{M}}$ commutes with $\boldsymbol{X}^{\otimes N}$ and $\boldsymbol{H}_{\mathrm{P}}$ commutes with $\boldsymbol{Z}^{\otimes N}$. But we could find more symmetries by decomposing the phase Hamiltonian as follows:

$$\begin{aligned} \boldsymbol{H}_{\mathrm{P}} &= \sum_{k=1}^{\lfloor K/2 \rfloor} \boldsymbol{F}_{2k} + \sum_{k=1}^{\lfloor K/2 \rfloor} \boldsymbol{F}_{2k-1} \\ &:= \boldsymbol{H}_{\mathrm{P}}^{(\mathrm{even})} + \boldsymbol{H}_{\mathrm{P}}^{(\mathrm{odd})}. \end{aligned} \tag{7.29}$$

We note that

*Remark* 7.7. The "even part" $\boldsymbol{H}_{\mathrm{P}}^{(\mathrm{even})}$ commutes with $\boldsymbol{X}^{\otimes N}$, while the "odd part" $\boldsymbol{H}_{\mathrm{P}}^{(\mathrm{odd})}$ anti-commutes with $\boldsymbol{X}^{\otimes N}$. Furthermore, $e^{-\imath\gamma_n \boldsymbol{H}_{\mathrm{P}}^{(\mathrm{even})}}$ also commutes with both $\boldsymbol{X}^{\otimes N}$ and $\boldsymbol{Z}^{\otimes N}$, since $[\boldsymbol{A},\boldsymbol{B}] = 0$ implies $[e^{\imath\theta \boldsymbol{A}},\boldsymbol{B}] = 0$.

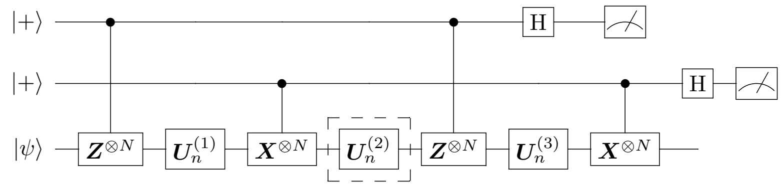


FIGURE 7.20: Circuit implementing a single stage of the QAOA on an even number of qubits $N$, protected by two STS checks. The gates in the dashed boxes are protected from any single-qubit error by the STS checks.

To see this more clearly, let us consider the QUBO case (7.24), for which we have

$$\begin{aligned} \boldsymbol{H}_{\mathrm{P}}^{(\mathrm{even})} &= \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}\boldsymbol{Z}_i\boldsymbol{Z}_j, \\ \boldsymbol{H}_{\mathrm{P}}^{(\mathrm{odd})} &= \sum_{i=1}^{N} b_i\boldsymbol{Z}_i, \end{aligned} \tag{7.30}$$

where $a_{ij}$ denotes the $(i,j)$-th entry of $\boldsymbol{A}$ and $b_i$ denotes the $i$-th entry of $\boldsymbol{b}$. Observe that the operator $\boldsymbol{Z}_i\boldsymbol{Z}_j$ commutes with $\boldsymbol{X}^{\otimes N}$, while $\boldsymbol{Z}_i$ anti-commutes with $\boldsymbol{X}^{\otimes N}$. In general, any Pauli operator constituted by the tensor product of an even number of Pauli-Zs would commute with $\boldsymbol{X}^{\otimes N}$, whereas it would anti-commute with $\boldsymbol{X}^{\otimes N}$, if the number of Pauli-Z's is odd.

Since the gates implementing $e^{-\imath\gamma_n\boldsymbol{H}_{\mathrm{P}}}$ commute with one another, we may rearrange the order of execution of these gates, so that $e^{-\imath\gamma_n\boldsymbol{H}_{\mathrm{P}}^{(\mathrm{odd})}}$ is executed before $e^{-\imath\gamma_n\boldsymbol{H}_{\mathrm{P}}^{(\mathrm{even})}}$. This leads to the following decomposition of the $n$-th stage into three sub-stages

$$\begin{aligned} \boldsymbol{U}_n(\beta_n,\gamma_n) &= \boldsymbol{U}_n^{(3)}\boldsymbol{U}_n^{(2)}\boldsymbol{U}_n^{(1)} \\ &= e^{-\imath\beta_n\boldsymbol{H}_{\mathrm{M}}}e^{-\imath\gamma_n\boldsymbol{H}_{\mathrm{P}}^{(\mathrm{even})}}e^{-\imath\gamma_n\boldsymbol{H}_{\mathrm{P}}^{(\mathrm{odd})}}. \end{aligned} \tag{7.31}$$

This tripartite circuit has the following STSs

$$\boldsymbol{S}_1 = \mathcal{S}\{\boldsymbol{Z}^{\otimes N}(0),\boldsymbol{Z}^{\otimes N}(2)\},\ \ \boldsymbol{S}_2 = \mathcal{S}\{\boldsymbol{X}^{\otimes N}(1),\boldsymbol{X}^{\otimes N}(3)\}. \tag{7.32}$$

A noteworthy fact is that $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ are not simultaneously observable when $N$ is odd. Therefore, we arrive at different circuit implementations for even $N$ and odd $N$, as shown in Fig. 7.20 and 7.21, respectively. The ancillas can be re-initialized and reused in the subsequent stages. The main difference between the two implementations is that the third sub-stage $\boldsymbol{U}_n^{(3)}$ is not protected when $N$ is odd, and thus the circuits having odd $N$ and those having even $N$ are not equally protected. Fortunately, the third sub-stage only consists of single-qubit gates that are typically less noisy in practice. Also note that the second sub-stage $\boldsymbol{U}_n^{(2)} = e^{-\imath\gamma_n\boldsymbol{H}_{\mathrm{P}}^{(\mathrm{even})}}$ commutes with both $\boldsymbol{X}^{\otimes N}$ and $\boldsymbol{Z}^{\otimes N}$, hence we could detect any single-qubit error that occurs within this sub-stage.

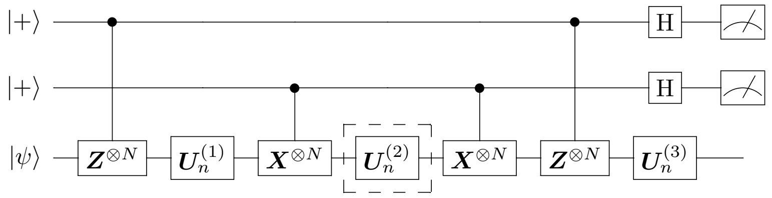


FIGURE 7.21: Circuit implementing a single stage of the QAOA on an odd number of qubits $N$, protected by two STS checks. The gates in the dashed boxes are protected from any single-qubit error by the STS checks.

## 7.5 Numerical Results

In this section, we characterize the performance of the STS method using numerical examples. When evaluating the computational accuracy, we use the purity[3] of the output state of the data register as the performance metric, defined by $\mathrm{Tr}\left\{\rho_{\mathrm{data}}^2\right\}$, where $\rho_{\mathrm{data}}$ is the output state of the data register.

### 7.5.1 Consecutive Single-Qubit Gates

We first contrast the STS method to the quantum switch based method described in Section 7.2, using the low-complexity example of single-qubit circuits. Specifically, we consider consecutive X-rotation gates applied to a single qubit. Since the gates are diagonal under the X-basis, we do not expect that any of the two methods would detect X-errors. In light of this, we assume that each X-rotation gate is associated with a Pauli-Z (dephasing) channel having the error probability of $\epsilon_1 = 0.001$. The two-qubit gates applied in both error mitigation methods are also assumed to be contaminated by Pauli-Z errors at an error probability of $\epsilon_2$. We will consider different values of $\epsilon_2$ in the following discussion.

Let us first consider the case of $\epsilon_2/\epsilon_1 = 2$. This is an idealistic case for quantum switches, since the controlled rotation gates (e.g. the gate $A$ in Fig. 7.2) inflict an error on the data register at the same probability as that of the uncontrolled gates (e.g. the gate $B$ in Fig. 7.2). However, this is typically not the case for practical devices, for which $\epsilon_2/\epsilon_1$ is around 10. We portray the simulation results in Fig. 7.22 where we have $N_{\mathrm{G}} = 2$ consecutive X-rotation gates, while in Fig. 7.23 we have $N_{\mathrm{G}} = 10$. As we have discussed in Section 7.2.1, there are multiple possible implementations of the quantum switch based method, when $N_{\mathrm{G}} > 2$. In Fig. 7.23, “quantum switch, type-1” refers to the implementation shown in Fig. 7.3, while “quantum switch, type-2” refers to that shown in Fig. 7.4.

---

[3] Instead of evaluating directly the error of certain computational tasks, we use the purity because it does not depend on the specific observable, and hence may reflect the performance of the error mitigation techniques more clearly.

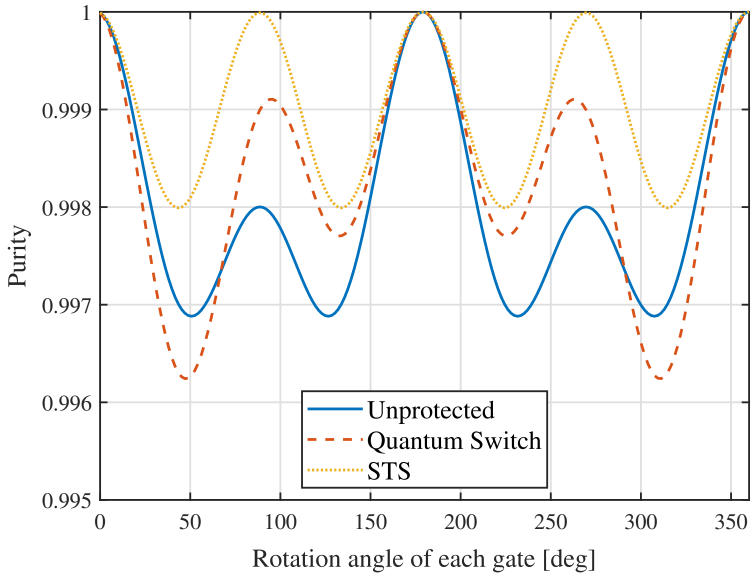


FIGURE 7.22: The output purities of different implementations of two consecutive X-rotation gates vs. the rotation angle of each gate, where $\epsilon_2/\epsilon_1 = 2$.

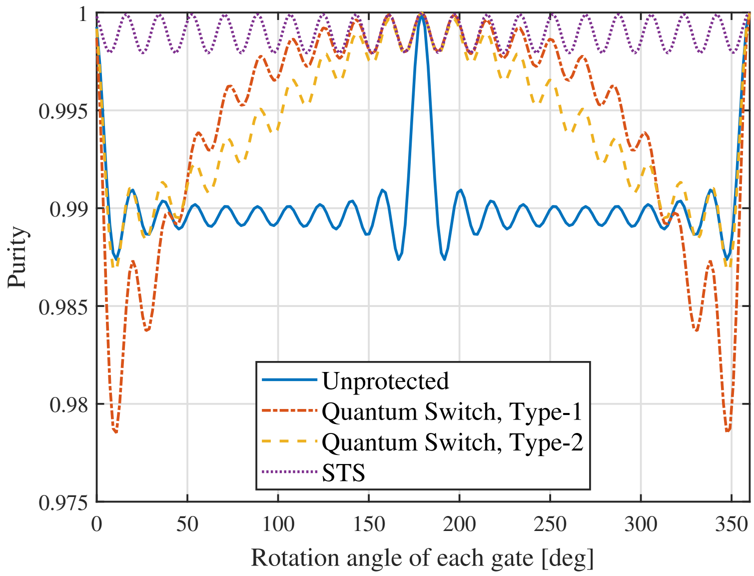


FIGURE 7.23: The output purities of different implementations of ten consecutive X-rotation gates vs. the rotation angle of each gate, where $\epsilon_2/\epsilon_1 = 2$.

Observe from Fig. 7.22 that the output purity of both the quantum switch and of the STS depends on the rotation angle of each X-rotation gate. To elaborate, the rotation angle has an impact on the commutativity with the Z-error, which in turn determines the error mitigation performance. Observe from Fig. 7.23 that, compared to the unprotected circuits, the accuracy improvement of both methods becomes more significant when $N_G$ is larger, since the additional error introduced by the methods themselves becomes less severe than that of the consecutive X-rotations. An interesting phenomenon is that the quantum switch based method performs better for larger rotation angles.

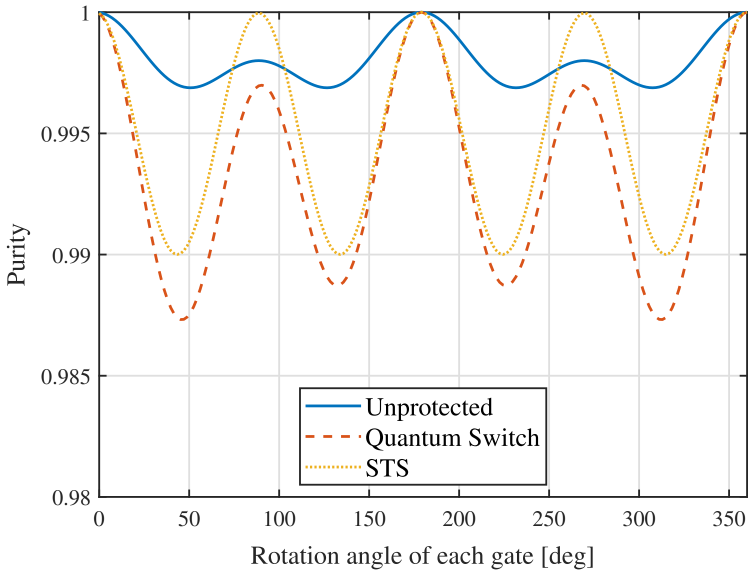


FIGURE 7.24: The output purities of different implementations of two consecutive X-rotation gates vs. the rotation angle of each gate, where $\epsilon_2/\epsilon_1 = 10$.

This may be interpreted as a penalty of treating the X-rotation gate itself as the reference of symmetry verification, instead of using a universal reference (e.g. Pauli-X operators in the STS method).

The results are portrayed for the more practical case of $\epsilon_2/\epsilon_1 = 10$ in Fig. 7.24 and 7.25. We see that the quantum switch based method is only beneficial for a limited range of rotation angles in the $N_G = 10$ case, while STS is beneficial across a wider range. Note that the STS technique may be generalized to more complex circuits. Hence may expect that STSs are potentially beneficial for a large range of practical circuits, while quantum switches might only be useful for certain special circuits. However, it is noteworthy that using STSs requires the knowledge of the specific type of symmetry, while quantum switches are applicable as long as we know that certain gates commute with each other.

### 7.5.2 QFT Circuits

In this subsection, we evaluate the error mitigation performance of STSs when applied to $N$-qubit QFT circuits.

Specifically, we consider the combined STS shown in Fig. 7.19. The output purities under various channel models are shown in Fig. 7.26. Observe that STSs are more beneficial under Y-error as well as X-error channels, and they are even detrimental for Z-error channels. This is as expected, since the STSs of QFT circuits commute with Z-errors. As for the sampling overhead, it is seen from Fig. 7.27 that the sampling overhead factor increases with the error detection probability, as may be inferred from its definition (7.21).

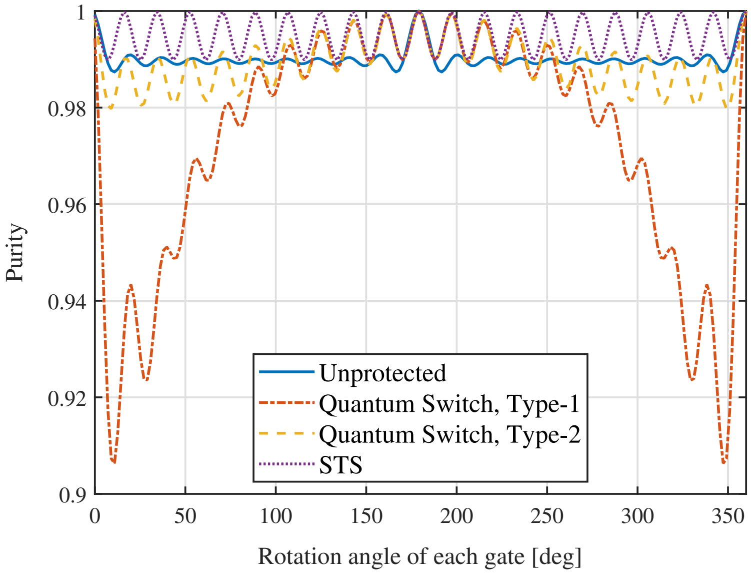


FIGURE 7.25: The output purities of different implementations of ten consecutive X-rotation gates vs. the rotation angle of each gate, where $\epsilon_2/\epsilon_1 = 10$.

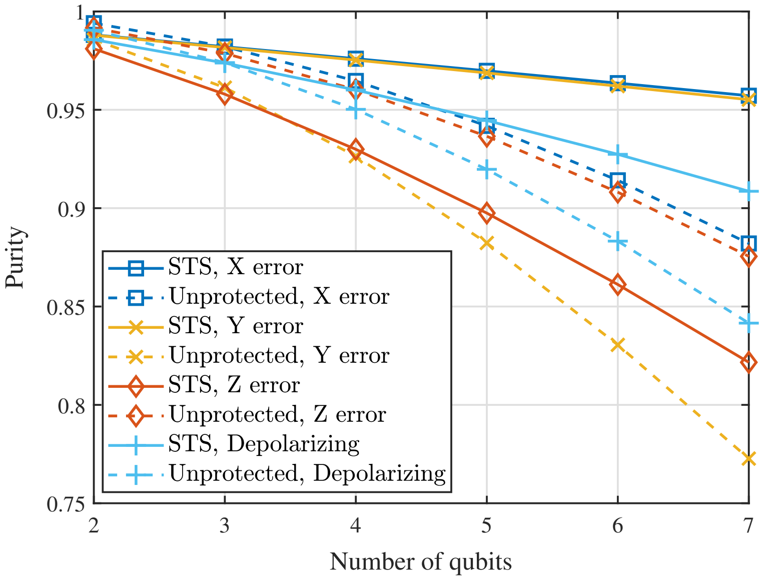


FIGURE 7.26: The output purities QFT circuits under different channel models, as functions of the number of qubits. The gate error rate is 0.003 for two-qubit gates, and 0.0003 for single-qubit gates.

The output purity versus the gate error rate under depolarizing channels is illustrated in Fig. 7.28. Here we consider the practical case of $\epsilon_2/\epsilon_1 = 10$, where $\epsilon_1$ and $\epsilon_2$ are the error rates of single-qubit and two-qubit gates, respectively. The curves marked by “STS, error-free check” correspond to the idealistic case where the gates used for implementing STS checks are error-free. We see that the purity decreases approximately linearly as the gate error rate increases. It is also noteworthy that the purity decreases faster for larger $N$, since the number of gates is also larger.

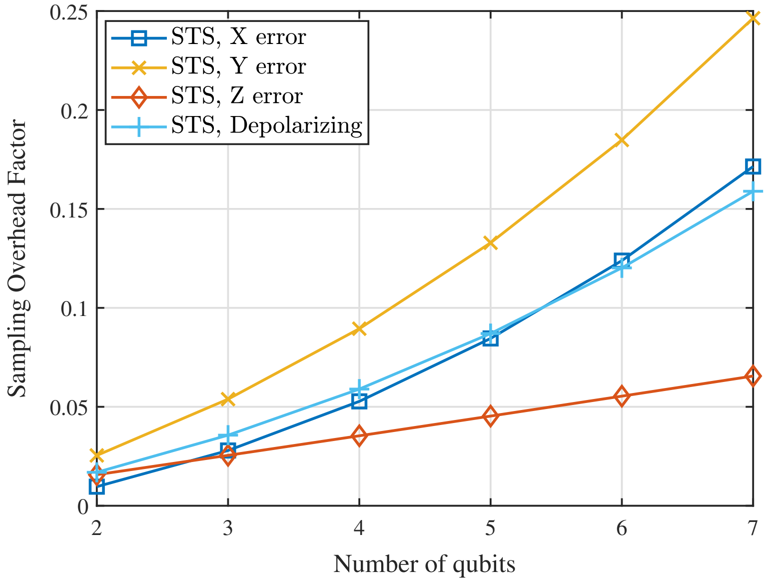


FIGURE 7.27: The sampling overhead factors of QFT circuits under different channel models, as functions of the number of qubits. The gate error rate is 0.003 for two-qubit gates, and 0.0003 for single-qubit gates.

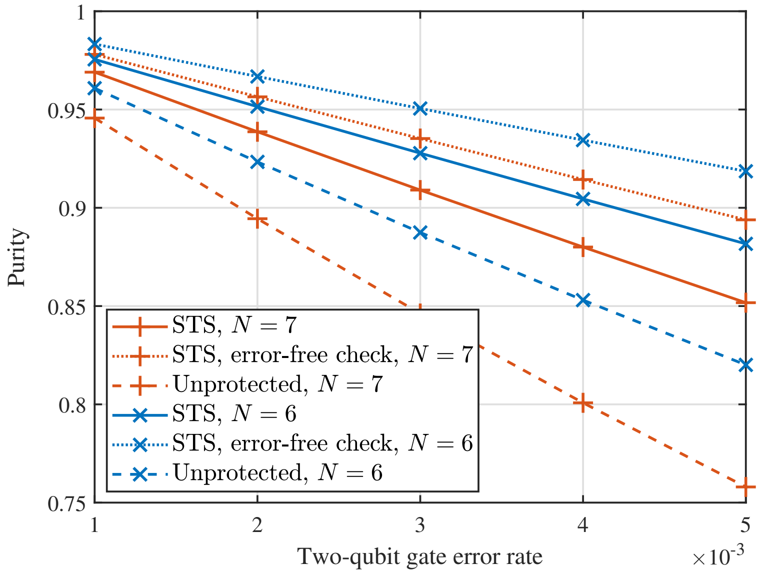


FIGURE 7.28: The output purities of QFT circuits under depolarizing channels, as functions of the error rate of two-qubit gates. The error rate of single-qubit gates is 1/10 that of two-qubit gates.

We conclude that, for QFT circuits, the STS method is particularly beneficial for asymmetric channels, for example, when the rate of X-errors is 10 times that of Z-errors. Note that the specific type of the error does not matter as long as the channel is asymmetric, because we may apply a global rotation to the entire circuit for ensuring that the dominant type of errors does not commute with the gates.

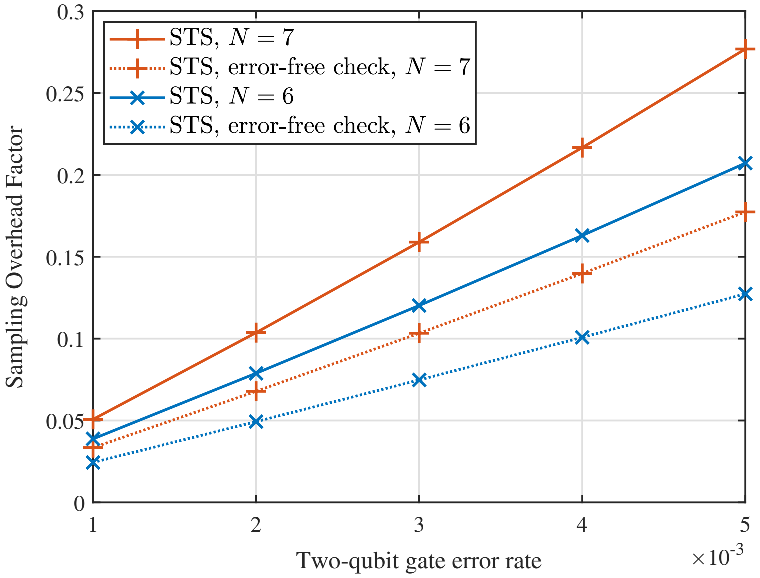


FIGURE 7.29: The sampling overhead factors of QFT circuits under depolarizing channels, as functions of the error rate of two-qubit gates. The error rate of single-qubit gates is 1/10 that of two-qubit gates.

### 7.5.3 QAOA Circuits

Finally, let us evaluate the performance of STSs applied to QAOA circuits discussed in Section 7.4.2 and 7.4.3. We first consider single-stage QAOA circuits, denoted as $\text{QAOA}_1$ circuits. For the simulations in this subsection, we use the following phase Hamiltonian

$$\boldsymbol{H}_{\text{P}} = \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}\boldsymbol{Z}_i\boldsymbol{Z}_j + \sum_{i=1}^{N} b_i\boldsymbol{Z}_i, \tag{7.33}$$

where $a_{ij}$ and $b_i$ are randomly drawn from the uniform distribution over the interval $(-1, 1)$. The simulation results are then averaged over 1000 random instances of the parameters. Every two-qubit gate is affected by a depolarizing channel having a depolarizing probability of 0.001, while the single-qubit gates have 10 times lower depolarizing probabilities.

The output purities and the sampling overhead factors are shown in Fig. 7.30 and 7.31, respectively. In these figures, "STS, $\text{Cat}_2$" refers to the implementation of STSs relying on cat states defined on two ancillas, as portrayed in Fig. 7.16. The specific implementation of QAOA circuits is portrayed in Fig. 7.32.

Observe from Fig. 7.30 that the STS method relying on cat states defined on two ancillas outperforms its counterpart relying on a single ancilla. This corroborates with our discussion on the mitigation of error proliferation in Section 7.3.4, and demonstrates the trade-off between accuracy and qubit overhead. The sampling overhead factors shown in Fig. 7.31 are on the order of the corresponding error detection probability, similar to our previous discussion on QFT circuits in Section 7.5.2.

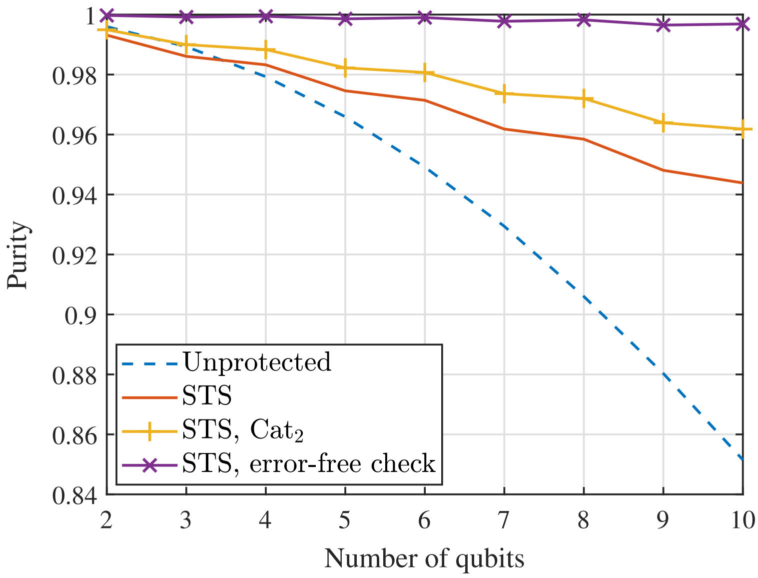


FIGURE 7.30: The output purities of different implementations of the QAOA$_1$ circuit, as functions of the number of qubits.

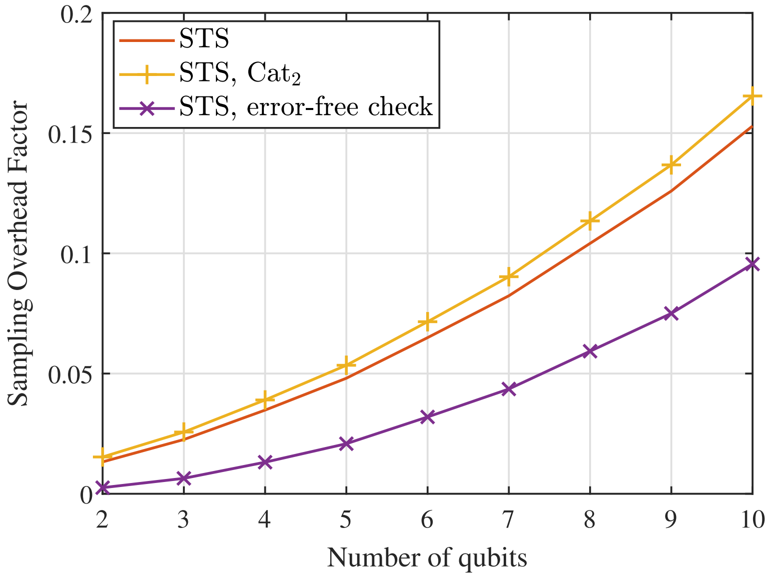


FIGURE 7.31: The sampling overhead factors of different implementations of the QAOA$_1$ circuit, as functions of the number of qubits.

Note that the purity curves of STS methods in Fig. 7.30 are not smooth. This is due to the fact that QAOA circuits relying on an even number of qubits and those on an odd number of qubits are not equally protected. Indeed, as we may observe from Fig. 7.20 and 7.21, the final sub-stage corresponding to the mixing Hamiltonian is not protected, when the number of data qubits $N$ is odd, which is due to the simultaneous observability issue of the STSs. Consequently, the purities of QAOA circuits having odd $N$ are lower than the expected purity, when the simultaneous observability is not an issue.

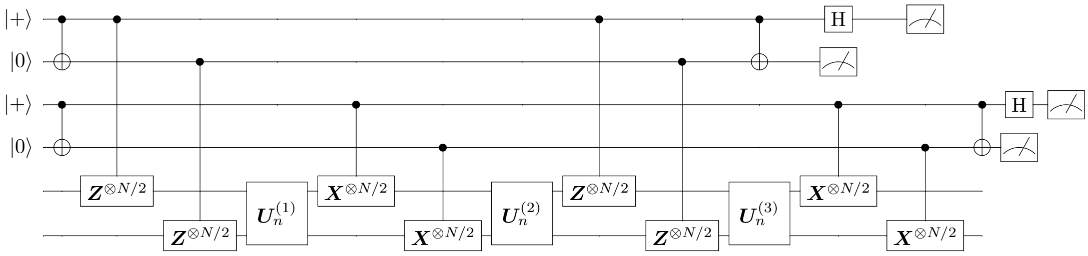


FIGURE 7.32: The implementation of an STS-protected $\text{QAOA}_1$ circuit, relying on two control qubits.

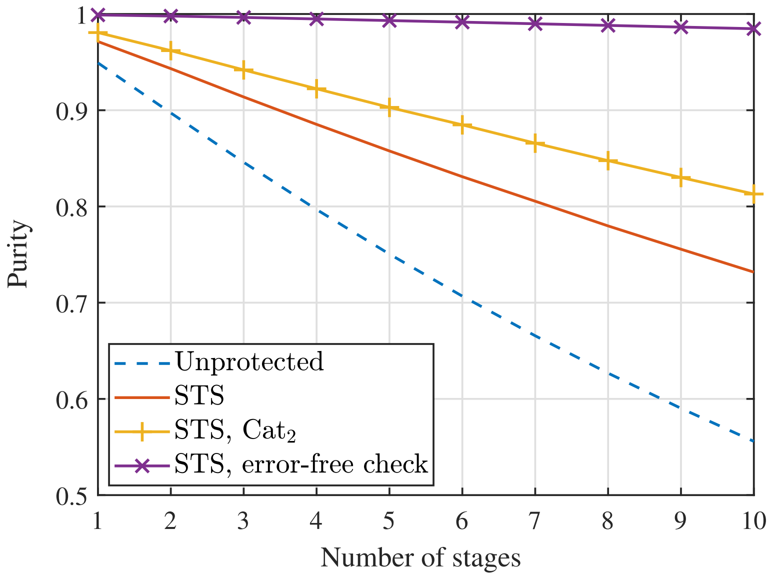


FIGURE 7.33: The output purities of different implementations of multistage QAOA circuits, as functions of the number of stages.

Next we consider multistage QAOA circuits. The components of the parameter vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are randomly drawn from uniform distributions on $(-\pi, \pi)$. As it can be seen from Fig. 7.33, the purity of the cat-state STS method decreases more slowly than that of the STS method relying on a single ancilla. Due to the complexity escalation of emulating quantum circuits on classical computers, we cannot produce the results of the STS method relying on larger cat states defined on $N_c > 2$ ancillas. We conjecture that the purity can be further improved by using more ancillas, which is ultimately upper-bounded by the purity when the gates used for STS checks are error-free.

## 7.6 Conclusions and Future Research

### 7.6.1 Conclusions

In this chapter, we have proposed a general framework for circuit-oriented symmetry verification. Specifically, the quantum switch based method can be directly applied,

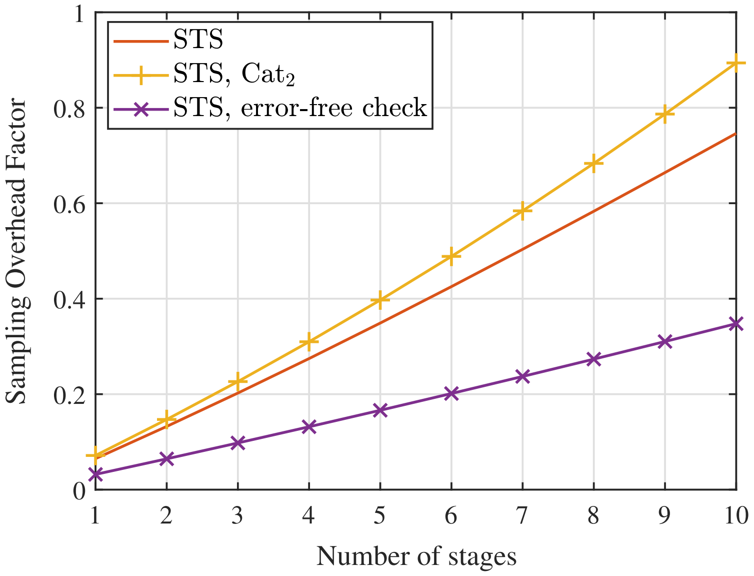


FIGURE 7.34: The sampling overhead factors of different implementations of multi-stage QAOA circuits, as functions of the number of stages.

when certain gates are known to commute with each other. For the case where the circuit has known symmetries, we propose the method of STS, generalizing the concept of conventional stabilizers used for state-oriented symmetry verifications. This method is capable of verifying the symmetries without the knowledge of the current quantum state. Another major difference between STSs and their conventional counterparts is that they are not necessarily simultaneously observable, and hence sometimes a re-arrangement of the circuit is required to perform multiple STS checks. We have also discussed the accuracy vs. overhead trade-off of STSs, and provided quantum circuit designs that strike flexible trade-offs. Finally, we have demonstrated the performance of the proposed methods using numerical examples concerning practical quantum algorithms, including the QFT and the QAOA.

### 7.6.2 Future Research

- As discussed in Section 7.3.4, the additional circuitry introduced by STSs would cause error proliferation among data qubits, which may be mitigated by splitting a single STS into multiple simultaneously observable STSs. In conventional state-oriented symmetry verification, the circuitry overhead-induced error proliferation can be suppressed to its minimum by relying on post-selection [36]. Specifically, instead of using ancillas and controlled gates to perform non-destructive measurements, we may measure the stabilizers by computing the weighted sum over multiple outcomes of destructive measurements. For example, if we know (prior to the computation) that the state $\rho$ at the output of the circuit is stabilized by a stabilizer $\mathcal{S}$, we may compute the expectation of some operator $\mathcal{M}$ as follows:

$$\mathrm{Tr}\left\{\mathcal{M}\rho\right\} = \frac{\mathrm{Tr}\left\{\mathcal{M}\rho\right\} + \mathrm{Tr}\left\{\mathcal{M}\mathcal{S}\rho\right\}}{1 + \mathrm{Tr}\left\{\mathcal{S}\rho\right\}}. \tag{7.34}$$

This would significantly reduce the circuitry overhead, since the only overhead comes from the basis transformation required by the destructive measurements. However, this method seems unlikely to be extendible to STSs. To elaborate, STSs are properties of a part of a quantum circuit, which do not necessarily hold at the output of the circuit, which the destructive measurements take place. By contrast, state symmetries are preserved by the entire circuit, hence can be naturally verified using destructive measurements. Therefore, we conjecture that STSs do not have efficient post-selection-based implementations. Proving or disproving this conjecture would significantly improve our understanding of STSs from both theoretical and practical perspectives.

- Another possible future research direction is to find more practical algorithms for which STSs is beneficial. In the "treasury house" of quantum algorithms, one of the algorithms that seem to be most suitable for STSs is the Hamiltonian simulation discussed in Section 2.5.1. Specifically, the Hamiltonians may themselves satisfy certain symmetry conditions. For example, when a Hamiltonian $\mathcal{H}$ can be decomposed as

  $$\mathcal{H} = \sum_{k=1}^{K} \theta_k \mathcal{H}_k, \tag{7.35}$$

  where $\mathcal{H}_k,\ k = 1 \ldots K$ are sub-Hamiltonians that may be readily implemented using native quantum gates, the symmetry of these sub-Hamiltonians may be exploited for error mitigation. Since the sub-Hamiltonians do not necessarily commute with each other, the symmetries are most likely to be circuit symmetries instead of state symmetries. Thus we may apply our STS method to mitigate errors that violates the symmetries. In this context, an important research topic is to determine the best quantum circuit compilation, in which the symmetries may be maximally exploited.

# Chapter 8

# Algorithm-Level Error Mitigation for Quantum Phase Estimation

## 8.1 Introduction

We have established in Chapter 3 that, besides circuit-level errors caused by decoherence and hardware imperfections, another major source of errors in quantum computation is constituted by algorithm-level errors, which is rooted in the fact that quantum algorithms are of probabilistic nature, and are typically characterized based on approximate formulas. Notably, algorithm-level errors cannot be corrected by QECCs, hence we have to resort to QEM methods. In Section 3.4, we have discussed techniques that mitigate the Trotterization error [75] of Hamiltonian simulation. In this chapter, we investigate the algorithm-level error mitigation in the context of quantum phase estimation [91, 150, 151], which is another key enabler of quantum computational speedup over classical computers.

In particular, quantum phase estimation is a widely used component of quantum algorithms providing substantial acceleration over their best classical counterpart, including Shor's factoring algorithm [12], the Harrow-Hassidim-Lloyd (HHL) algorithm [16] conceived for solving linear systems, and the quantum counting algorithm [94, 201]. Recently, iterative quantum phase estimators have also been proposed for potential application in near-term NISQ computers [202–204].

The circuit diagram of quantum phase estimation is portrayed in Fig. 8.1, where the initial state $|\psi\rangle$ of the data register is an eigenstate of the unitary operator $\mathcal{U}$ satisfying $\mathcal{U}|\psi\rangle = e^{j\varphi}|\psi\rangle$, and $\mathrm{QFT}_M^{-1}$ denotes the inverse quantum Fourier transform [48, Sec. 5.1] applied to $M$ qubits, which is a quantum-domain version of the classical discrete Fourier transform [205, Sec. 8]. This algorithm aims for estimating the phase $\varphi$. Broadly speaking, the $2^M - 1$ controlled unitaries produce a $2^M$-point sinusoidal signal in the

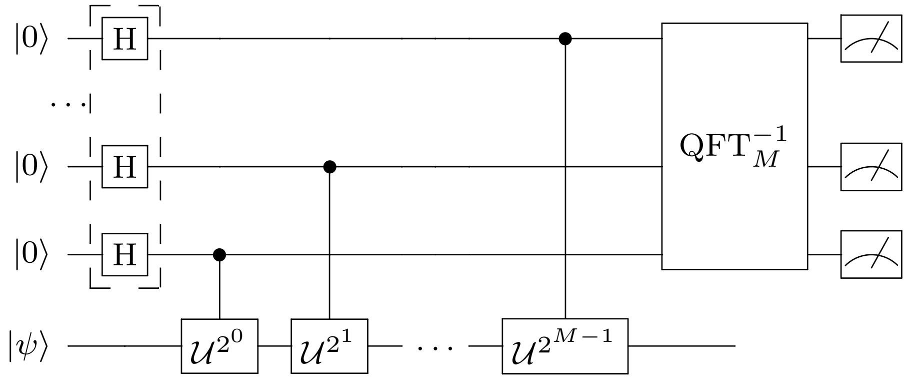


FIGURE 8.1: The circuit diagram of quantum phase estimation.

control register, whose frequency corresponds exactly to the desired phase $\varphi$. This may be viewed as the quantum-domain version of the power method of eigendecomposition [133]. When efficient implementations of the power of the unitary operator $\mathcal{U}$ are available, exponential speedup over classical algorithms is possible, as seen in Shor's algorithm [12].

Quantum phase estimation yields the exact result, when the recording length $2^M$ is an integer multiple of the period of the sinusoidal signal [48, Sec. 5.2.1]. When this is not the case, the spectral leakage problem [206] arises, which is an important topic in classical signal processing methods related to the discrete Fourier transform. The quantum-domain version of this problem is somewhat more grave, since the measurement outcome would assume erroneous values at non-negligible probabilities, hence the information about the correct phase is lost.

One of the conventional solutions to the spectral leakage issue is to multiply the time-domain signal by a smooth "window function" [206]. This idea has been applied to the quantum phase estimation problem in [207], where the authors show that the cosine window [205, Sec. 7] is optimal in terms of the mean squared error (MSE) of single-sample-based estimation. An efficient quantum circuit-based implementation of the cosine window has later been proposed in [208], also showing that it has satisfactory performance in terms of the error probability.

While existing treatises focus on single-sample-based estimation, in this chapter, we consider the spectral leakage mitigation attained by multiple samples (i.e. measurement outcomes). This is motivated by the fact that the coherence time of NISQ computers is limited [31], hence classical computing power may be harnessed, by appropriately fusing multiple samples. The rest of this chapter is organized as follows. We first formulate the spectral leakage problem and discuss the existing countermeasures in Section 8.2. Next, we derive the CRB of the windowed quantum phase estimation in Section 8.3, and propose the dual-frequency estimator in Section 8.4. The numerical results are then presented in Section 8.5, and we conclude in Section 8.6.

## 8.2 The Spectral Leakage Problem and Windowing Methods

Let us commence by deriving of the output state of the quantum circuit used for phase estimation and shown in Fig. 8.1. After the Hadamard gates, the joint quantum state of both the control register and the target register becomes

$$|+\rangle^{\otimes M}|\psi\rangle = \frac{1}{\sqrt{2^M}}\sum_{n=0}^{2^M-1}|n\rangle|\psi\rangle, \tag{8.1}$$

which will be referred to as the *initial state* of the quantum phase estimation algorithm in the rest of this chapter. The subsequent controlled-$\mathcal{U}$ gates transform the initial state into the following state

$$\begin{aligned}|\phi\rangle &= \frac{1}{\sqrt{2^M}}\sum_{n=0}^{2^M-1}|n\rangle\,\mathcal{U}^n|\psi\rangle \\ &= \frac{1}{\sqrt{2^M}}\bigotimes_{k=0}^{M-1}\Big(|0\rangle + e^{j2^k\varphi}|1\rangle\Big)|\psi\rangle \\ &= \frac{1}{\sqrt{2^M}}\sum_{n=0}^{2^M-1}e^{jn\varphi}|n\rangle|\psi\rangle.\end{aligned} \tag{8.2}$$

Note that the inverse quantum Fourier transform $\mathrm{QFT}_M^{-1}$ may be expressed as follows[1]

$$\mathrm{QFT}_M^{-1} = \frac{1}{\sqrt{N}}\sum_{m=0}^{N-1}e^{-j\frac{2\pi m}{N}}|m\rangle\langle n|, \tag{8.3}$$

where $N = 2^M$ is the record length. Hence the quantum state of the control register at the output of the $\mathrm{QFT}_M^{-1}$ may be expressed as

$$|\phi\rangle_{\text{out}} = \frac{1}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}e^{jn\left(\varphi-\frac{2\pi m}{N}\right)}|m\rangle. \tag{8.4}$$

The probability of observing the outcome $|y\rangle$ is thus given by

$$f(y;\varphi) = |\langle y|\,\phi\rangle_{\text{out}}|^2 = \frac{1}{N^2}\left|\sum_{n=0}^{N-1}e^{jn\left(\varphi-\frac{2\pi y}{N}\right)}\right|^2. \tag{8.5}$$

When $\varphi = 2\pi k/N$, $k \in \mathbb{Z}$, we see that the probability $f(y;\varphi)$ takes the maximum value of 1 at $y = k$, hence the phase estimation yields the exact solution. However, when $\varphi$ is not an integer multiple of $2\pi/N$, $f(y;\varphi)$ is non-zero for almost all $0 \le y \le N-1$,

---

[1] In classical signal processing theory, the transform having the minus sign "−" in the exponents is typically referred to as the discrete Fourier transform, and the transform having the plus sign is referred to as the inverse discrete Fourier transform. However, the quantum computing community is using a different convention, which refers the transform (8.3) as the inverse quantum Fourier transform [48, Section 5.1].

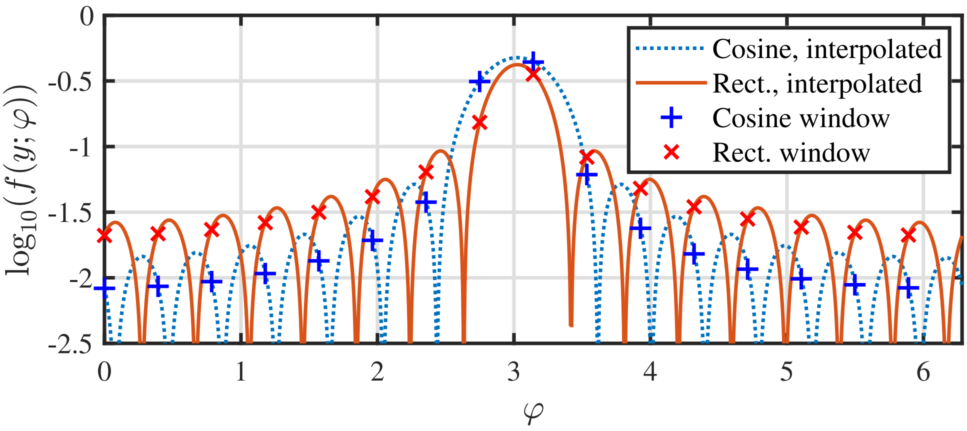


FIGURE 8.2: Illustration of the observation probability distribution $f(y;\varphi)$ for rectangular and cosine windows, respectively.

causing large estimation errors. This phenomenon is referred to as "spectral leakage" in the literature of classical signal processing [206, 209].

The spectral leakage issue can be mitigated using the classic windowing method [210], multiplying the input signal by a "window" function defining a weighting vector $\boldsymbol{\alpha}$. This results in the following observation probability distribution

$$f(y;\varphi) = \frac{1}{N}\left|\sum_{n=0}^{N-1} \alpha_n e^{jn\left(\varphi-\frac{2\pi y}{N}\right)}\right|^2, \tag{8.6}$$

where $\alpha_n$ is the $n$-th entry of $\boldsymbol{\alpha}$, and $\|\boldsymbol{\alpha}\| = 1$. Upon comparing (8.5) to (8.6), we see that the original quantum phase estimation corresponds to a rectangular window of $\boldsymbol{\alpha} = \frac{1}{\sqrt{N}}\mathbf{1}$. This weighting procedure may be implemented upon replacing the Hadamard gates in the dashed box of Fig. 8.1 by customized state preparation circuits [208].

It has been shown that in terms of the lowest MSE, the optimal window is the cosine window [207, 211], given by

$$\alpha_n^{(\cos)} = \sqrt{\frac{2}{N}} \sin\left[\frac{\pi(n-1)}{N}\right]. \tag{8.7}$$

The authors of [208] have designed an efficient state preparation circuit for the cosine window. The effect of the window may be intuitively interpreted by considering the corresponding observation probability distribution, as portrayed in Fig. 8.2. Observe that the sidelobes of the cosine window are much lower than those of the rectangular window, hence the probability that extremely large errors occur is substantially reduced. However, it is also seen that the cosine window has a wider mainlobe. If the sidelobes can be suppressed without widening the mainlobe, we may achieve better performance than that of the cosine window. This motivates our dual-frequency estimator, which will be detailed in Section 8.4.

## 8.3 CRB Analysis

In this section, we derive the CRB of the quantum phase estimation problem, which will be used as a performance benchmark in the numerical simulations of Section 8.5. It also motivates us to propose the dual-frequency estimator detailed in Section 8.4, which asymptotically approaches the CRB upon increasing the number of samples.

The CRB is a lower bound on the mean-squared error of estimators [212], which is tight, when the noise is weak or the number of observations is large. In light of this, it may be used as the metric for determining the optimal window function for a large number of samples.

In general, given the likelihood function $f(y;\theta)$ of an observation $y$, the CRB of the parameter $\theta$ is given by

$$\mathbb{E}\{(\theta-\hat{\theta})^2\} \geq \frac{1}{\mathrm{FI}(\theta)}, \tag{8.8}$$

where

$$\mathrm{FI}(\theta) = \mathbb{E}\left\{\left[\frac{\partial \ln f(y;\theta)}{\partial\theta}\right]^2\right\}. \tag{8.9}$$

is the Fisher information of $\theta$ [212]. When there are multiple independent and identical distributed observations, the total Fisher information is the sum of the Fisher information of all individual observations.

In the context of quantum phase estimation, the likelihood function of a single observation is given by (8.6), which may be rewritten in a more compact form as

$$\begin{aligned} f(y;\varphi) &= \frac{1}{N}|\boldsymbol{e}_y^{\mathrm{H}}\boldsymbol{\alpha}|^2 \\ &= \frac{1}{N}\boldsymbol{e}_y^{\mathrm{H}}\boldsymbol{\alpha}\boldsymbol{\alpha}^{\mathrm{H}}\boldsymbol{e}_y, \end{aligned} \tag{8.10}$$

where

$$\boldsymbol{e}_y := \left[1,\ \exp\left\{j\left(\varphi-\frac{2\pi y}{N}\right)\right\}, \ldots,\ \exp\left\{j(N-1)\left(\varphi-\frac{2\pi y}{N}\right)\right\}\right]^{\mathrm{T}}$$

denotes the vector containing all the phase terms in (8.5).

For the simplicity of further derivation, we rewrite the Fisher information in (8.9) in the following alternative form

$$\begin{aligned}
\mathrm{FI}(\varphi) &= \mathbb{E}\left\{\left[\frac{\partial \ln f(y;\theta)}{\partial\theta}\right]^2\right\} \\
&= \sum_y f(y;\theta)\left[\frac{\partial \ln f(y;\theta)}{\partial\theta}\right]^2 \\
&= \sum_y \frac{1}{f(y;\varphi)}\cdot\left(\frac{\partial}{\partial\varphi}f(y;\varphi)\right)^2,
\end{aligned} \tag{8.11}$$

where the partial derivative may be expressed as

$$\begin{aligned}
\frac{\partial}{\partial\varphi}f(y;\varphi) &= \frac{1}{N}\left[\left(\frac{\partial \boldsymbol{e}_y}{\partial\varphi}\right)^{\mathrm{T}}\boldsymbol{\alpha}^*\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{e}_y^* + \left(\frac{\partial \boldsymbol{e}_y^*}{\partial\varphi}\right)^{\mathrm{T}}\boldsymbol{\alpha}\boldsymbol{\alpha}^{\mathrm{H}}\boldsymbol{e}_y\right] \\
&= \frac{j}{N}(\boldsymbol{e}^T\boldsymbol{N}\boldsymbol{\alpha}^*\boldsymbol{\alpha}^T\boldsymbol{e}^* - \boldsymbol{e}^H\boldsymbol{N}\boldsymbol{\alpha}\boldsymbol{\alpha}^H\boldsymbol{e}) \\
&= \frac{j}{N}\boldsymbol{e}_y^{\mathrm{H}}(\boldsymbol{\alpha}\boldsymbol{\alpha}^{\mathrm{H}}\boldsymbol{N} - \boldsymbol{N}\boldsymbol{\alpha}\boldsymbol{\alpha}^{\mathrm{H}})\boldsymbol{e}_y,
\end{aligned} \tag{8.12}$$

where we have

$$\boldsymbol{N} = \mathrm{diag}\{0, 1, \ldots, N-1\}.$$

Upon introducing $\boldsymbol{A} = \boldsymbol{\alpha}\boldsymbol{\alpha}^{\mathrm{H}}$, the Fisher information may be expressed as

$$\begin{aligned}
\mathrm{FI}(\varphi) &= \frac{1}{N}\sum_{y=0}^{N-1}\frac{|\boldsymbol{e}_y^{\mathrm{H}}(\boldsymbol{\alpha}\boldsymbol{\alpha}^{\mathrm{H}}\boldsymbol{N} - \boldsymbol{N}\boldsymbol{\alpha}\boldsymbol{\alpha}^{\mathrm{H}})\boldsymbol{e}_y|^2}{\boldsymbol{e}_y^{\mathrm{H}}\boldsymbol{\alpha}\boldsymbol{\alpha}^{\mathrm{H}}\boldsymbol{e}_y} \\
&= \frac{1}{N}\sum_{y=0}^{N-1}\frac{|\boldsymbol{e}^H[\boldsymbol{A},\boldsymbol{N}]\boldsymbol{e}|^2}{\boldsymbol{e}^H\boldsymbol{A}\boldsymbol{e}} \\
&= \frac{4}{N}\sum_{y=0}^{N-1}\frac{\mathrm{Im}^2\{\boldsymbol{e}_y^{\mathrm{H}}\boldsymbol{A}\boldsymbol{N}\boldsymbol{e}_y\}}{\boldsymbol{e}_y^{\mathrm{H}}\boldsymbol{A}\boldsymbol{e}_y}.
\end{aligned} \tag{8.13}$$

When there are $N_{\mathrm{s}}$ samples, due to their mutual independence, the total Fisher information is simply formulated as $N_{\mathrm{s}} \cdot \mathrm{FI}(\varphi)$. This implies that the optimal window in the asymptotic regime of infinite samples may actually be determined by considering the Fisher information of the window applied to a single sample.

We may use (8.13) to obtain an intuition about the optimal window in the asymptotic regime. In Fig. 8.3, we compare the square-root CRB of various windows, where we set the record length to $N = 128$, corresponding to 7 control qubits. Observe that the rectangular window (equivalent to no windowing over a finite interval) corresponds to the lowest CRB. This indicates that it should have the best asymptotic performance. However, we see that the naive sample-mean-based estimator operates far from the CRB. By contrast, the sample-mean estimators of cosine and Bartlett windows exhibit near-CRB performances. We will further address this issue in Section 8.4.

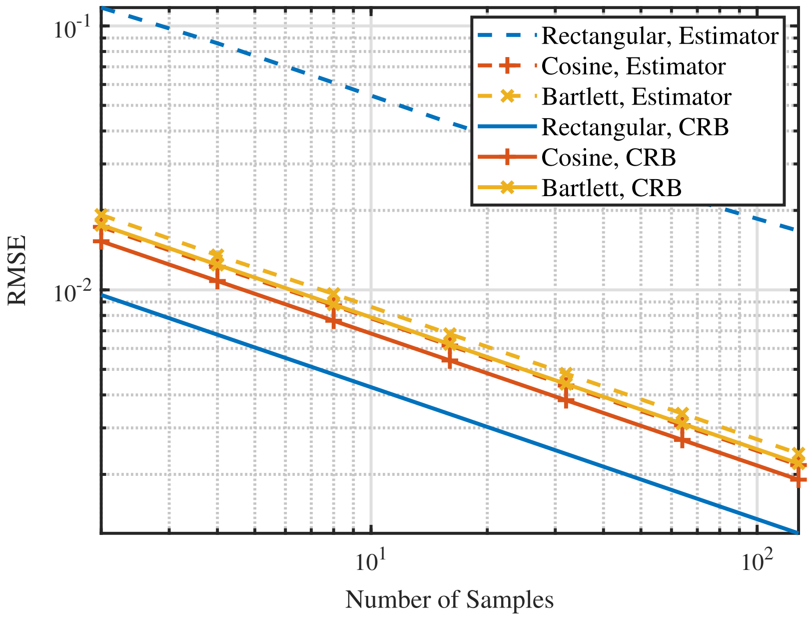


FIGURE 8.3: The square-root CRB vs. number of samples of various window functions, compared with the RMSE of corresponding sample-mean estimators.

## 8.4 The Dual-Frequency Estimator

Let us commence our discussion from constructing an approximate maximum likelihood estimator based on the rectangular window with the aid of $N_{\mathrm{s}}$ samples. We first denote the samples as $\boldsymbol{y} = [y_1, y_2, \ldots, y_{N_{\mathrm{s}}}]^{\mathrm{T}}$. From (8.5), the exact maximum likelihood estimator is given by

$$\hat{\varphi}_{\mathrm{ML}} = \underset{\varphi}{\operatorname{argmax}} \sum_{i=1}^{N_{\mathrm{s}}} \ln \left| \sum_{n=0}^{N-1} e^{jn\left(\varphi - \frac{2\pi y_i}{N}\right)} \right|^2 . \tag{8.14}$$

Since the samples only take integer values between 0 and $N-1$, we may use an alternative parametrization of $\boldsymbol{z} \in \mathbb{R}^N$, whose $n$-th entry $z_k$ represents the number of times that $n-1$ occurs in the samples $\boldsymbol{y}$. Thus we may rewrite (8.14) as

$$\hat{\varphi}_{\mathrm{ML}} = \underset{\varphi}{\operatorname{argmax}} \sum_{k=1}^{N} z_k \ln \left| \sum_{n=0}^{N-1} e^{jn\left(\varphi - \frac{2\pi k}{N}\right)} \right|^2 . \tag{8.15}$$

A naive strategy of solving (8.15) is an exhaustive search of $\varphi$ in $[0, 2\pi)$, which is computationally expansive. To simplify the problem, we first obtain a rough estimate of $\varphi$ as

$$\hat{\varphi}_{\mathrm{rough}} = \frac{2\pi}{N} \left[ \left( \underset{k}{\operatorname{argmax}}\, z_k \right) - 1 \right] . \tag{8.16}$$

Note that the worst-case complexity of this step is on the order of $O(N_{\mathrm{s}})$ when $N_{\mathrm{s}} \ll N$. Next we conduct a refined search within the interval of $[\hat{\varphi}_{\mathrm{rough}} - 2\pi/N, \hat{\varphi}_{\mathrm{rough}} +$

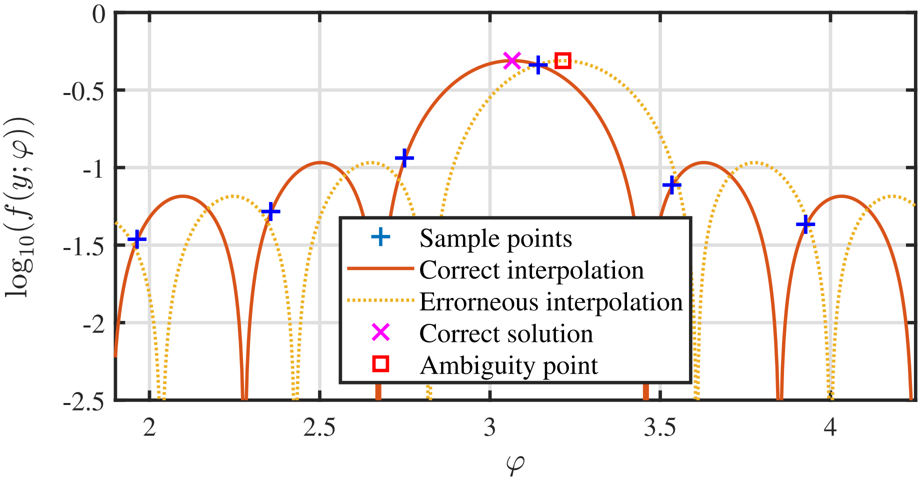


FIGURE 8.4: Illustration of the ambiguity problem associated with the approximate maximum likelihood estimator when $\varphi$ is close to $2\pi k/N$, $k \in \mathbb{Z}$.

$2\pi/N]$. Although more sophisticated optimization techniques may yield more favorable accuracy-complexity trade-off, here we consider the simple approach of a uniform grid search over $O(\sqrt{N_s})$ grid points, inspired by the fact that the Fisher information is on the order of $O(N_s)$. To avoid the summation over $n$ in (8.15), we consider the following approximation for large $N$:

$$\frac{1}{N}\left|\sum_{n=0}^{N-1} e^{jn\left(\varphi-\frac{2\pi k}{N}\right)}\right| \approx \left|\operatorname{sinc}\left(\frac{N\varphi}{2\pi}-k\right)\right|. \tag{8.17}$$

Using the above approximation, the worst-case complexity of the grid search is reduced to $O(N_s^{1.5})$. We denote the final estimate as $\hat{\varphi}_{\text{AML}}$, and the correction term as $e_{\text{AML}} = \hat{\varphi}_{\text{AML}} - \hat{\varphi}_{\text{rough}}$.

This estimator, however, does not in general yield satisfactory performance. We may develop some further intuition concerning this issue by revisiting Fig. 8.2. Observe that most of the information about $\varphi$ is contained in the pair of sample points within the mainlobe. When $\varphi$ is near $(2\pi k+1)/N$, $k \in \mathbb{Z}$, the largest and the second largest entries in $z$ may be used as reliable estimates of the two sample points. However, when $\varphi$ is close to $2\pi k/N$, $k \in \mathbb{Z}$, it is likely that the second largest entry in $z$ corresponds to the first sidelobe, causing an unexpected estimation error. As portrayed in Fig. 8.4, the difficulty of distinguishing the highest sidelobe from the sample point in the mainlobe having smaller likelihood incurs an ambiguity problem for the maximum likelihood estimator.

To understand this ambiguity problem in a more precise manner, let us consider a concrete example. Specifically, we set the record length to $N = 100$, the number of samples

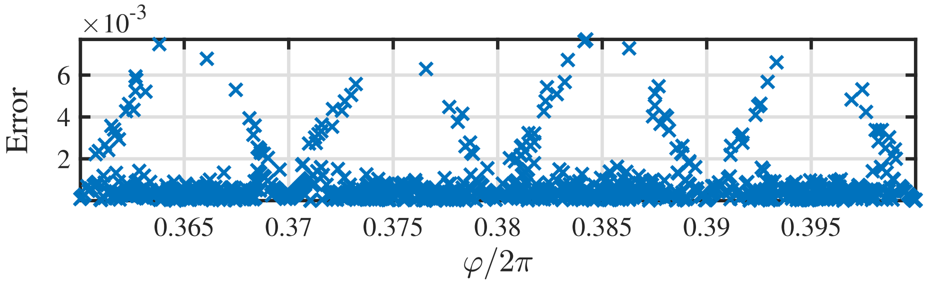


FIGURE 8.5: Scatter plot of the errors of the approximate maximum likelihood estimator for $\varphi$ uniformly distributed over a fixed interval.

to $N_\text{s} = 30$, and then plot the corresponding estimation error based on the approximate maximum likelihood estimator characterized in Fig. 8.5.[2] Observe that the errors are sometimes large when $\varphi$ is close to an integer multiple of $1/2\pi N$. In particular, the magnitude of the error grows linearly with the distance to the closest integer multiple of $1/2\pi N$. As we have discussed previously, this phenomenon originates from the fact that two possible interpolations of the sample points are hardly distinguishable, hence there is a non-zero probability that the estimator yields the erroneous result corresponding to the "mirror point" of the correct one across the line of $\varphi = k/2\pi N$, where $k$ is the closest integer multiple of $1/2\pi N$.

To tackle this problem, we split the samples into two sets, each having $N_\text{s}/2$ samples. For the first set, we compute the aforementioned approximate maximum likelihood estimator, and obtain $\hat{\varphi}_\text{rough,1}$ and $e_\text{AML,1}$. For the second set, we apply a frequency offset of $1/2N$ (one half of a frequency resolution unit) to the control register, resulting in the following maximum likelihood problem

$$\hat{\varphi}_\text{ML,2} = \frac{\pi}{N} + \underset{\varphi}{\operatorname{argmax}} \sum_{k=1}^{N} z_k \ln \left| \sum_{n=0}^{N-1} e^{jn\left(\varphi - \frac{2\pi k}{N} + \frac{\pi}{N}\right)} \right|^2 . \tag{8.18}$$

Similarly, we may obtain $\hat{\varphi}_\text{rough,2}$ and $e_\text{AML,2}$. Based on these results, we construct four candidate intermediate estimates, contained in a vector as follows:

$$\begin{aligned} \boldsymbol{u} = \Big[ & \hat{\varphi}_\text{rough,1} + e_\text{AML,1},\ \hat{\varphi}_\text{rough,1} - e_\text{AML,1}, \\ & \hat{\varphi}_\text{rough,2} + e_\text{AML,2},\ \hat{\varphi}_\text{rough,2} - e_\text{AML,2} + \frac{2\pi}{N} \Big]^\text{T} . \end{aligned} \tag{8.19}$$

Finally, we find the pair of entries in $\boldsymbol{u}$ that are the closest to each other, meaning that

$$(i, j) = \underset{(i,j)}{\operatorname{argmax}}\ (u_i - u_j)^2, \tag{8.20}$$

---

[2] In practice, the record length can only be an integer power of 2. Here we choose $N = 100$ only for the purpose of a clearer illustration.

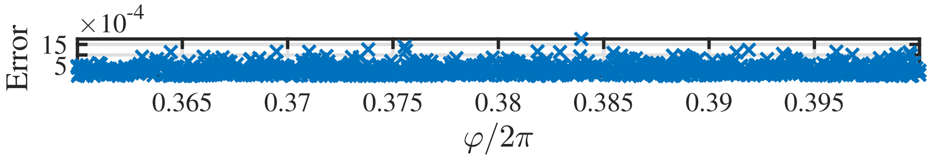


FIGURE 8.6: Scatter plot of the errors of the dual-frequency estimator for $\varphi$ uniformly distributed over a fixed interval.

and the final estimate is given by

$$\hat{\varphi}_{\mathrm{DF}} = \frac{1}{2}(u_i + u_j). \tag{8.21}$$

The overall complexity of the dual-frequency estimator is at most $O(N_{\mathrm{s}}^{1.5})$, which is comparable to the $O(N_{\mathrm{s}})$ complexity of the simple sample-mean estimator. In practice, the complexity can be lower than $O(N_{\mathrm{s}}^{1.5})$. To elaborate, note that the actual complexity of the dual-frequency estimator is proportional to

$$C_{\mathrm{DF}} = N_{\mathrm{bin}} \sqrt{N_{\mathrm{s}}}, \tag{8.22}$$

where $N_{\mathrm{bin}}$ denotes the number of non-empty "bins" $z_k \neq 0$. In fact, most of the bins would be empty, since the probability of observing those outcomes would be extremely low. Moreover, the bins containing a small number of samples would provide less information about the desired phase $\varphi$, since they are much noisier than the bins containing more samples. For the numerical simulations presented in this chapter, we choose 8 bins having the largest number of samples. In light of this, the actual complexity of the dual-frequency estimator becomes $O(N_{\mathrm{s}} + \sqrt{N})$, where the $O(N_{\mathrm{s}})$ comes from the fact that re-organizing the samples into bins would require $O(N_{\mathrm{s}})$ operations.

Next we show that the dual-frequency estimator indeed resolves the ambiguity problem using a concrete example. Similar to Fig. 8.5, we set the record length to $N = 100$ and the number of samples to $N_{\mathrm{s}} = 30$. The estimation error of the dual-frequency estimator is portrayed in Fig. 8.6. As seen from the figure, the dual-frequency estimator only produces small errors, and the linearly increasing trend of errors seen in Fig. 8.5 does not appear. This implies that the ambiguity point becomes distinguishable from the correct solution.

The rationale of the dual-frequency estimator may be intuitively understood as follows. To circumvent the difficulty of distinguishing the correct solution from the ambiguity point, we include both points into the set of candidate estimates, and also apply the same technique to the set of samples associated with the $(1/2N)$-frequency offset. Since the ambiguity points of both sets are unlikely to be the same due to the frequency offset, we may then identify the correct solution by finding the matching pair of candidate solutions using (8.20).

Finally, we show that the $(1/2N)$-frequency offset exploited in the estimator may be implemented using single-qubit phase rotation gates, hence the computational overhead of state preparation is negligible. Note that this offset may be obtained by initializing the input state of the control register as

$$|\phi\rangle_{\text{in}} = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} e^{\frac{j\pi n}{N}} |n\rangle . \tag{8.23}$$

This state admits the following simple tensor-product form

$$|\phi\rangle_{\text{in}} = \bigotimes_{m=1}^{M} \frac{1}{\sqrt{2}} \left( |0\rangle + e^{\frac{j\pi 2^{m-1}}{N}} |1\rangle \right), \tag{8.24}$$

which may be implemented by applying a phase rotation gate $\mathcal{R}_{\text{z}}\left(\frac{\pi 2^{m-1}}{N}\right)$ to each $m$-th qubit in the control register. This method bears some resemblance with the controlled-$\mathcal{U}$ gates in the quantum phase estimation circuit, which also constructs a sinusoidal signal (in other words, a frequency shift) on the control register.

## 8.5 Numerical Results

In this section, we characterize the performance of the proposed estimator using numerical simulations. We first consider the relationship between the RMSE of the estimators and the number of samples. In this example, we set the number of control qubits to $M = 7$, corresponding to $N = 128$. The number of samples $N_{\text{s}}$ varies from 2 to 100. The unknown phase $\varphi$ is randomly chosen from $(0, 2\pi)$ in each of the $10^5$ Monte Carlo trails. The RMSE of the dual-frequency estimator as well as that of the sample-mean estimator based on the cosine window is portrayed in Fig. 8.7, where the corresponding CRBs are also incorporated as benchmarks.

Observe that all curves in Fig. 8.7 exhibit the same linear trend in the asymptotic regime on the log-log scale. This implies that both estimators have the same asymptotic error scaling as the CRB, which scales on the order of $O(1/\sqrt{N_{\text{s}}})$ (in terms of RMSE). When $N_{\text{s}}$ is insufficiently large, the dual-frequency estimator exhibits a performance similar to that of the sample-mean estimator. But as $N_{\text{s}}$ increases, the performance of the dual-frequency estimator “switches” to near-CRB operation. Observe furthermore that it outperforms the cosine window-based method for $N_{\text{s}} \geq 16$. The relatively low accuracy of the dual-frequency estimator in the small-$N_{\text{s}}$ regime originates from the fact that the number of samples is to small for us to construct any reliable candidate estimate (as given in (8.19)). Consequently, the sample points that are far away from the

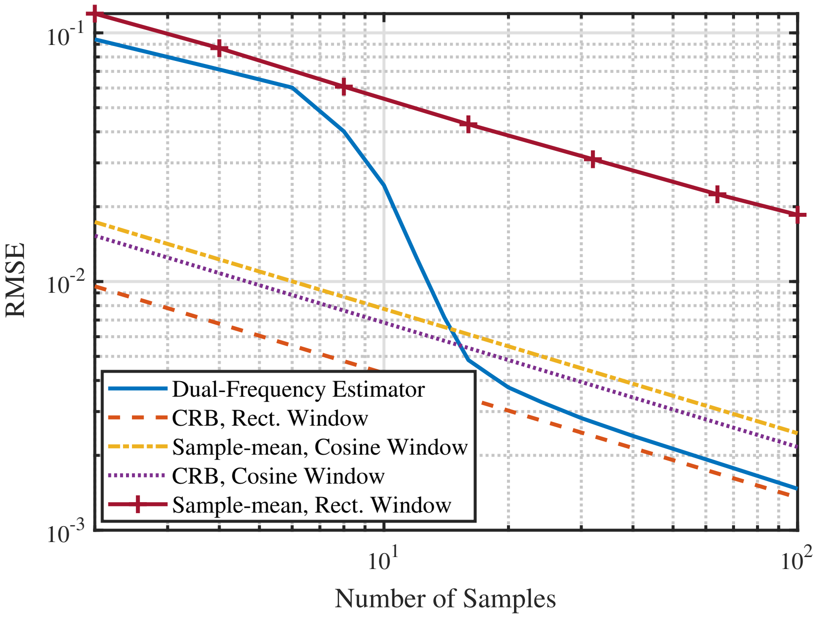


FIGURE 8.7: The phase estimation RMSE vs. the number of samples of both the proposed dual-frequency estimator and the cosine window method, compared to the corresponding CRBs.

true value of $\varphi$ (i.e., the "outliers") cannot be reliably identified, hence would cause large estimation errors.

Next we consider the dependence of the RMSE on the record length $N$. We set the number of samples to $N_s = 30$. The number of control qubits varies from 6 to 10, corresponding to $N = 64, 128, 256, 512, 1024$. The corresponding results are portrayed in Fig. 8.8. We also incorporate the approximate maximum likelihood estimator in Fig. 8.8 for a better illustration.

Observe that the sample-mean estimator based on the rectangular window exhibits an $O(1/\sqrt{N})$ scaling, while the others exhibit $O(1/N)$ scaling. This is a phenomenon that has also been observed in [207], which suggests that the rectangular window does not provide a substantial quantum speedup in the sense of RMSE scaling, since the $O(1/N)$ scaling (i.e., the "Heisenberg limit [93]) is an important characteristic of quantum algorithms conceived for phase estimation.

We also observe that the dual-frequency estimator does not exhibit the aforementioned bimodal phenomenon with respect to $N$. This suggests that for large $N$, we may still use a constant number of samples (for example, $N_s \geq 20$ as indicated by Fig. 8.7) to achieve a near-CRB performance.

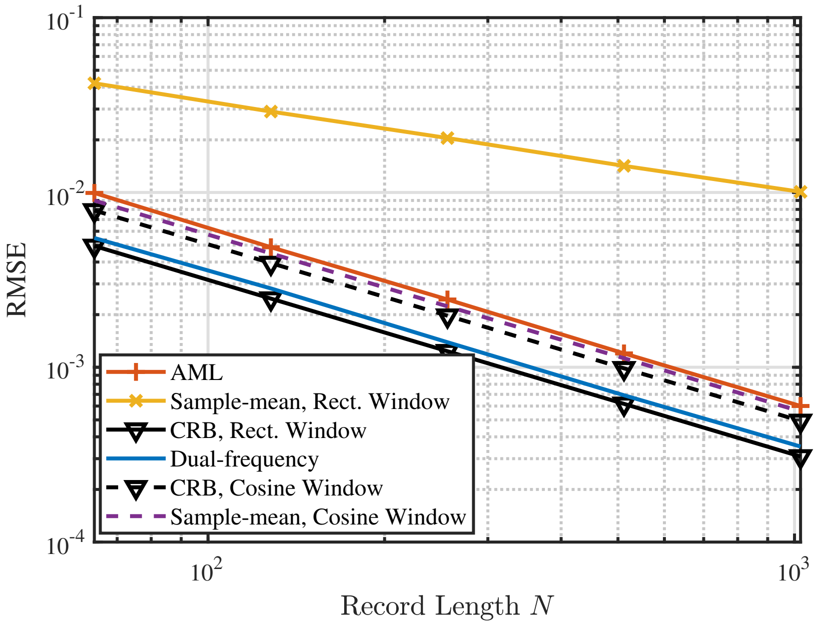


FIGURE 8.8: The phase estimation RMSE vs. the record length $N$ of the proposed dual-frequency estimator, the approximate maximum likelihood estimator, and the cosine window method, compared to the corresponding CRBs.

## 8.6 Conclusions and Future Research

### 8.6.1 Conclusions

In this chapter, a dual-frequency phase estimator was proposed for mitigating the spectral leakage-induced error in quantum phase estimation algorithm based on multiple samples. This is potentially useful in the context of noisy intermediate-scale quantum computing, since the maximum achievable recording length $N$ is ultimately restricted by the coherence time, hence we have to rely on multiple samples when a high phase estimation accuracy is required. To provide a systematic perspective, we have presented a CRB analysis of the quantum phase estimation problem in the asymptotic regime, which inspires the design of our dual-frequency estimator. Compared to the naive sample-mean estimator, the proposed estimator attains the Heisenberg limit in the sense of exhibiting an RMSE scaling on the order of $O(1/N)$. Furthermore, the estimator is capable of outperforming the cosine window, which is shown to be optimal for single-sample estimation, when the number of samples is sufficiently large (but constant with respect to $N$).

### 8.6.2 Future Research

- When quantum phase estimation serves as the final subroutine in the entire computational task, our method is straightforwardly applicable. For example, in the quantum counting algorithm aiming for finding the number of entries satisfying certain conditions in a database, the quantum phase estimation algorithm is exploited to estimate the speed of amplitude accumulation as the quantum amplitude amplification subroutine being repeatedly executed. The speed is then read out from the quantum register, which is proportional to the desired number of entries. In this example, we may directly use our dual-frequency estimator to enhance the accuracy of the estimation.

- Our method might also inspire future research on the algorithmic error mitigation for a broader range of quantum algorithms, especially those tailored for noisy-intermediate scale quantum computers. For example, in the original design of variational quantum algorithms, the eigenvalues are directly read out using Pauli measurements [62]. Although it is easy to implement, this procedure has an unfavorable computational complexity scaling, which is on the order of $O(1/\epsilon^2)$, where $\epsilon$ is the desired accuracy. By contrast, one may use quantum phase estimation to achieve a complexity scaling of $O(1/\epsilon)$, but at the cost of increased circuit depth. The authors of [213] proposed to use multiple samples in quantum phase estimation, in order to strike a beneficial trade-off between overall computational complexity and the circuit depth. In this context, our dual-frequency estimator may be applied to improve this trade-off, in the sense that a lower complexity may be achieved using the same circuit depth.

# Chapter 9

# Conclusions

In this thesis, the design of QEM has been improved. Explicitly, we have established that QEM may be applied for mitigating both circuit-level and algorithm-level errors. To strike a beneficial accuracy vs. overhead trade-off, we have developed several novel QEM techniques, summarized as follows:

- Regarding channel inversion based QEM aiming for mitigating circuit-level errors, we have presented a comprehensive sampling overhead analysis, and proposed a computationally efficient Monte Carlo implementation.
- For another class of circuit-level error mitigation techniques, namely for symmetry-based QEM, we have proposed the method of permutation filtering inspired by classical single processing theory, which is a flexible QEM technique relying on permutation symmetry. Furthermore, we have also proposed the technique of STS, which extends the conventional symmetry verification technique from state symmetries to circuit symmetries.
- Finally, we have proposed an algorithm-level error mitigation method, termed as the dual-frequency estimator, for improving the accuracy of the quantum phase estimation algorithm.

More specifically, our results in this thesis may be summarized in a chapter-by-chapter manner as follows:

- In **Chapter 1**, we have presented a brief overview of error-resilient quantum computation, including early research efforts focusing on fault-tolerant quantum computation and recent advances in the area of hybrid quantum-classical computation. In particular, we have highlighted the major milestones of hybrid quantum-classical computation as well as quantum error mitigation in Section 1.2.2. Our novel contributions in this thesis are then summarized in Section 1.3, and the organization of the thesis is presented in Section 1.4.

- In **Chapter 2**, we have introduced the preliminaries of quantum computation and quantum information. Specifically, in Section 2.1, we have introduced the fundamental unit of quantum information processing, namely the qubit, and contrasted it to its classical counterpart. We then elaborated the states of qubits and multi-qubit systems in Section 2.2 and Section 2.3, followed by a discussion on quantum channels modelling the imperfections of quantum gates in Section 2.4. In Section 2.5, we have discussed the design philosophy of representative quantum algorithms, including "conventional" quantum algorithms relying on the assumption of fault-tolerance in Section 2.5.1, and hybrid quantum-classical algorithms in Section 2.5.2. In particular, the relationship between these algorithms is portrayed in Fig. 9.1.

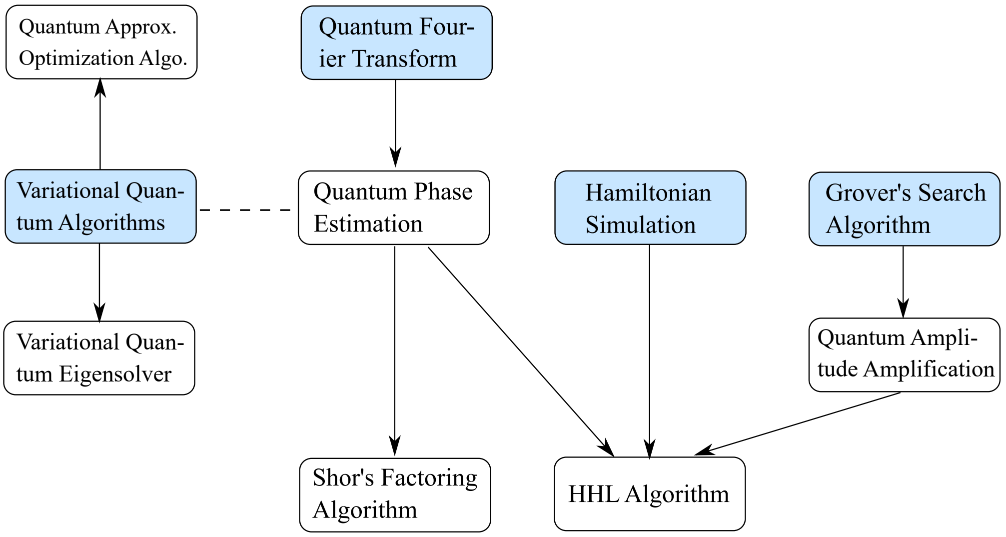


FIGURE 9.1: The relationship between the representative quantum algorithms introduced in Chapter 2. Coloured boxed represent the fundamental subroutines. Arrows imply that the succeeding algorithm is developed based on the preceding algorithm, while dashed lines connect algorithms that solve the same problem.

- In **Chapter 3**, we have presented an overview of QEM, outlining the taxonomy of QEM techniques, as portrayed in Fig. 9.2. The difference between circuit-level and algorithm-level error sources was highlighted in Section 3.1. For circuit-level errors, we have discussed the conventional technique of QECCs and QEDCs, which typically require a substantial qubit overhead. Then, we introduced circuit-level QEM techniques in Section 3.3, including channel inversion based QEM in Section 3.3.1, symmetry-based QEM in Section 3.3.2, zero-noise extrapolation based QEM in Section 3.3.3, and learning-based QEM in Section 3.3.4. For algorithm-level errors, we have introduced the techniques mitigating the error of Trotter approximation in Section 3.4. In particular, we highlighted that zero-noise extrapolation, a circuit-level QEM technique, may also be applied for mitigating algorithm-level errors.
- In **Chapter 4**, we have presented a comprehensive analysis of the sampling overhead factor of CI-QEM under various channel conditions. We first discussed the

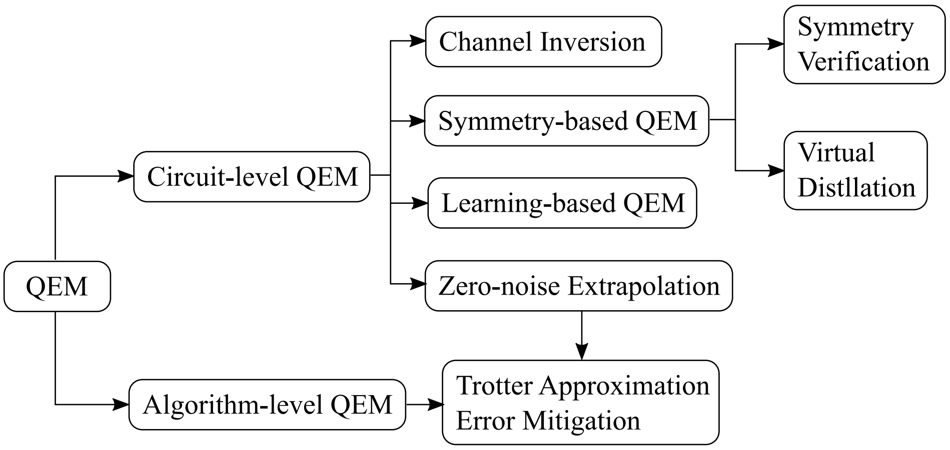


FIGURE 9.2: The taxonomy of existing QEM techniques introduced in Chpater 3.

quality metrics of quantum channels in Section 4.1, where we proposed the metric of GGEP as a natural metric for our sampling overhead analysis. For uncoded gates affected by errors modelled by general CPTP channels, we have shown in Section 4.2.2 that Pauli channels have the lowest SOF among all triangular channels (which includes the amplitude damping channels) having the same GGEP. Following this line of reasoning, in Section 4.2.3, we have shown furthermore that depolarizing channels have the lowest SOF in the family of all Pauli channels. These analytical results are illustrated in Fig. 9.3.

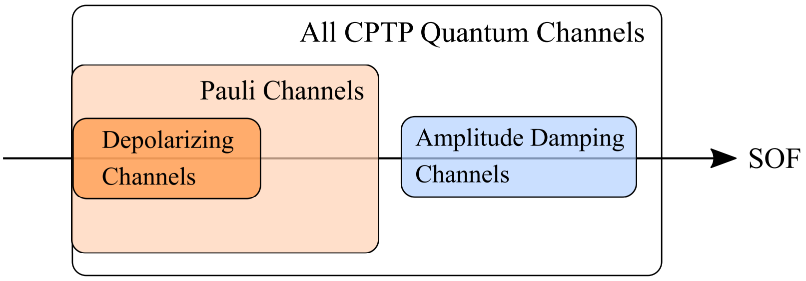


FIGURE 9.3: Illustrative comparison between the SOFs of different quantum channels having the same GGEP, according to the discussions in Chapter 4.

In Section 4.3, we conceived the QECC-CI-QEM as well as the QEDC-CI-QEM schemes, and have shown that there exist several critical quantum circuits sizes, beyond which sophisticated codes having more concatenation stages is more preferable, and vice versa. Specifically, for QEDC-CI-QEM, we have demonstrated in Section 4.3.4 that it may not be compatible with the popular transversal gate configuration, but they may still have beneficial applications, when the logical gates are appropriately designed.

- In **Chapter 5**, we have analyzed the error scaling versus sampling overhead trade-off of a Monte Carlo based implementation of CI-QEM aiming for reducing the computational cost of the candidate circuit generation for channel inversion. To understand the practicality of the proposed implementation, we have investigated the trade-off between the computational overhead and the error scaling behaviour of both quantum circuits protected by Monte Carlo based CI-QEM and their non-QEM-protected counterparts. The main analytical results in this chapter are summarized in Table 9.1.

TABLE 9.1: The characteristics of different error mitigation strategies discussed in Chapter 5.

| Performance Metric | **Monte Carlo Based** | Exact Inversion | No QEM |
|---|---|---|---|
| Error scaling vs. the number of gates $N_\mathrm{G}$ | $O(\sqrt{N_\mathrm{G}})$ | $O(\sqrt{N_\mathrm{G}})$ | $O(N_\mathrm{G})$ |
| Error scaling vs. the gate error rate $\epsilon$ | $O(\sqrt{\epsilon})$ | Unknown | $O(\epsilon)$ |
| Error scaling vs. the number of samples $N_\mathrm{s}$ | $O(\sqrt{N_\mathrm{s}})$ | $O(\sqrt{N_\mathrm{s}})$ | N/A |
| Candidate circuit generation complexity | $O(N_\mathrm{s}N_\mathrm{G})$ | $O\{\exp(N_\mathrm{G})\}$ | 0 |

As for the non-QEM-protected circuits, we have shown in Section 5.3.2 that the dynamic range of the noisy computational results shrinks exponentially as the number of gates $N_\mathrm{G}$ increases, implying a linear error scaling with $N_\mathrm{G}$. By contrast, in Section 5.3.4, we have shown that the error scales as the square root of $N_\mathrm{G}$ in the presence of Monte Carlo-based QEM, at the same computational cost as that of CI-QEM based on exact channel inversion. Moreover, the error scaling of Monte Carlo based QEM can be further improved at the cost of an increased computational complexity. Then, in Section 5.4, we have presented an intuitive geometric interpretation of the error scaling of Monte Carlo based QEM, using the central limit theorem. Finally, in Section 5.5, we have illustrated the analytical results both for low-complexity examples and for a more practical example of the quantum approximate optimization algorithm employed for multi-user detection in wireless communications.

- In **Chapter 6**, we have proposed a general framework for designing FIR-like permutation filters for mitigating the computational errors of variational quantum algorithms, relying on the permutation symmetry across different activations of quantum circuits. In particular, we define these filters in Section 6.2, which obey a polynomial form, combining different order of VD circuits. The spectral response of permutation filters is illustrated in Fig. 9.4. The original VD circuits may be viewed as special cases of permutation filters having only a single non-zero coefficient.

  We then proposed a filter design algorithm in Section 6.2.2, and proved that the filter design problem is an invariant convex (i.e. the so-called "invex" [192]) problem, hence the algorithm is guaranteed to converge to the global optimum. In Section 6.3, we have also proved a polynomial error reduction compared to VD for narrowband noise scenarios. This implies that permutation filters improve the

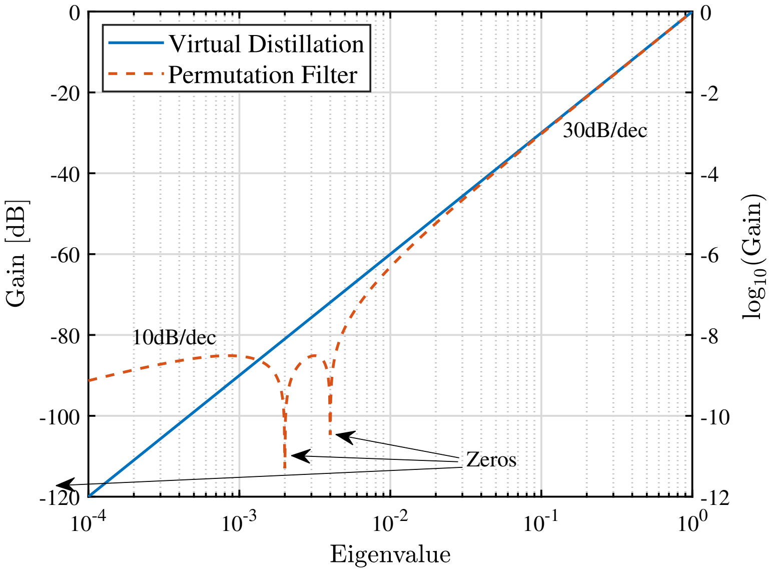
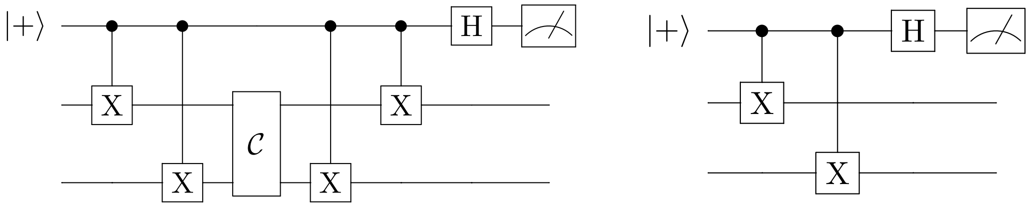


FIGURE 9.4: The spectral response of a third-order permutation filter proposed in Chapter 6, compared to that of the third-order VD.

error-reduction performance more substantially for quantum circuits having large depth or higher gate error rate.

- In **Chapter 7**, we have proposed a general framework for circuit-oriented symmetry verification, inspired by the quantum switch based method originally proposed in the quantum communications area for improving the capacity of quantum channels. When certain gates are known to commute with each other, we proposed in Section 7.2 to directly apply the quantum switch method to mitigate the errors violating the commutativity conditions. For the scenario where the circuit has known symmetries, in Section 7.3, we proposed the method of STS, generalizing the concept of conventional stabilizers used for state-oriented symmetry verifications. This method is capable of verifying the symmetries without the knowledge of the current quantum state. The quantum circuit implementation of STSs is contrasted to that of conventional stabilizers in Fig. 9.5.



(a) The STS $\mathcal{S}\{X_1(0), X_2(0), X_1(1), X_2(1)\}$.

(b) The conventional stabilizer $X_1X_2$.

FIGURE 9.5: Comparison between the STSs proposed in Chapter 7 and the conventional stabilizers.

We then discussed in Section 7.3.3 another major difference between STSs and their conventional counterparts, namely that they are not necessarily simultaneously observable, and hence sometimes a rearrangement of the circuit is required to perform multiple STS checks. We have also discussed the accuracy vs. overhead trade-off of STSs in Section 7.3.4, and provided quantum circuit designs that strike flexible trade-offs. To demonstrate the aforementioned design philosophy, in Section 7.4, we have provided specific design examples concerning a suite of practical quantum algorithms, including the QFT and the QAOA.

- In **Chapter 8**, we have proposed a dual-frequency phase estimator for mitigating the spectral leakage-induced algorithm-level error in quantum phase estimation algorithms based on multiple samples. This is potentially useful in the context of noisy intermediate-scale quantum computing, since the maximum achievable recording length $N$ is ultimately restricted by the coherence time, hence we have to rely on multiple samples, when a high phase estimation accuracy is required.

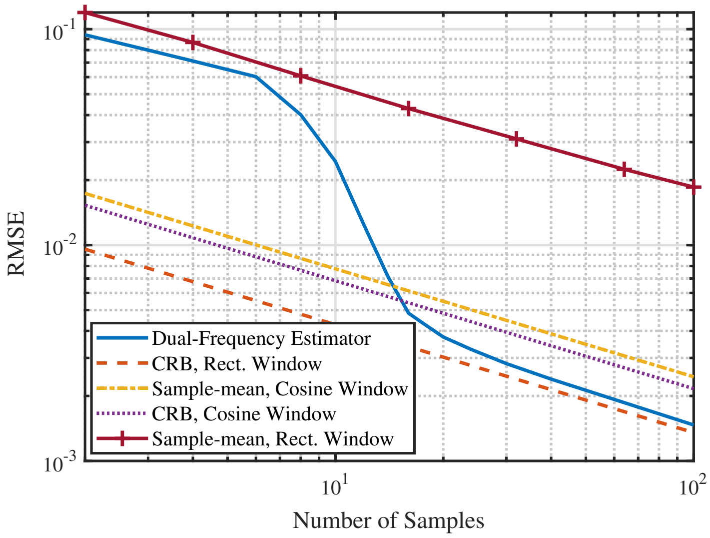


FIGURE 9.6: The phase estimation RMSE vs. the number of samples of both the dual-frequency estimator proposed in Chapter 8 and the cosine window method, compared to the corresponding CRBs.

To provide a systematic perspective, in Section 8.3, we have presented the CRB analysis of the quantum phase estimation problem in the asymptotic regime. Inspired by the CRB analysis, in Section 8.4, we proposed the concept of a dual-frequency estimator. Compared to the naive sample-mean estimator, the proposed estimator approaches the Heisenberg limit in the sense of exhibiting an RMSE scaling on the order of $O(1/N)$. Furthermore, the estimator is capable of outperforming the cosine window, which is shown to be optimal for single-sample estimation, when the number of samples is sufficiently large (but constant with respect to $N$), as portrayed in Fig. 9.6.

# Appendix A

# Proofs of Propositions

## A.1 Proof of Proposition 4.3

*Proof.* To prove our claim, it suffices to show that Pauli twirling does not increase the SOF. Without loss of generality, we assume that the specific columns corresponding to Pauli operators in $\boldsymbol{B}$ are the first $4^n$ columns. First we note that the quasi-probability representation corresponding to any CPTP channel having Pauli transfer matrix representation $\boldsymbol{C}$ may be expressed as:

$$\mu_{\boldsymbol{C}} = \boldsymbol{B}^{-1}\text{vec}\{\boldsymbol{C}^{-1}\}, \tag{A.1}$$

while the quasi-probability representation corresponding to the Pauli-twirled channel is given by

$$\mu_{\mathcal{T}_{\mathscr{P}}\boldsymbol{C}} = \boldsymbol{B}^{-1}\text{vec}\{[\mathcal{T}_{\mathscr{P}}\boldsymbol{C}]^{-1}\}, \tag{A.2}$$

where the super-operator $\mathcal{T}_{\mathscr{P}}$ represents the Pauli twirling operation. We can rewrite (A.2) in a matrix form as

$$\mu_{\mathcal{T}_{\mathscr{P}}\boldsymbol{C}} = \boldsymbol{B}^{-1}\text{vec}\{[\text{vec}^{-1}\{\boldsymbol{T}_{\mathscr{P}}\text{vec}\{\boldsymbol{C}\}\}]^{-1}\}, \tag{A.3}$$

where $\boldsymbol{T}_{\mathscr{P}}$ denotes the matrix representation of the Pauli twirling operator. Recall that the Pauli-twirled channel is given by

$$\mathcal{T}_{\mathscr{P}}\boldsymbol{C} = \frac{1}{4^n}\sum_{i=1}^{4^n}\mathcal{P}_i\boldsymbol{C}. \tag{A.4}$$

Upon introducing $\boldsymbol{B} = [\boldsymbol{b}_1\ \boldsymbol{b}_2\ \ldots\ \boldsymbol{b}_{16^n}]$, the Pauli operator $\mathcal{P}_i$ can be expressed in a matrix form as $\boldsymbol{P}_i = \boldsymbol{b}_i\boldsymbol{b}_i^{\text{T}}$. Thus the Pauli twirling operator can be represented as

$$\boldsymbol{T}_{\mathscr{P}} = \frac{1}{4^n}(\boldsymbol{B}\boldsymbol{I}_{\mathscr{T}})(\boldsymbol{B}\boldsymbol{I}_{\mathscr{T}})^{\text{T}}, \tag{A.5}$$

with $\boldsymbol{I}_{\mathscr{P}}$ being the following matrix

$$\boldsymbol{I}_{\mathscr{P}} = \begin{bmatrix} \boldsymbol{I}_{4^n} & \mathbf{0}_{4^n \times (16^n - 4^n)} \\ \mathbf{0}_{(16^n - 4^n) \times 4^n} & \mathbf{0}_{(16^n - 4^n) \times (16^n - 4^n)} \end{bmatrix}.$$

Thus we have

$$\boldsymbol{T}_{\mathscr{P}} \mathrm{vec}\{\boldsymbol{C}\} = \mathrm{vec}\{\mathrm{mdiag}\,\{\boldsymbol{C}\}\}.$$

Since the Pauli transfer matrix $\boldsymbol{C}$ is triangular, we have

$$(\mathrm{mdiag}\,\{\boldsymbol{C}\})^{-1} = (\mathrm{mdiag}\left\{\boldsymbol{C}^{-1}\right\}),$$

hence (A.3) can be simplified as

$$\mu_{\mathcal{T}_{\mathscr{P}}\boldsymbol{C}} = \boldsymbol{B}^{-1} \boldsymbol{T}_{\mathscr{P}} \mathrm{vec}\{\boldsymbol{C}^{-1}\}. \tag{A.6}$$

Using (A.1) and (A.6), we can show that if the statement of

$$\mu_{\mathcal{T}_{\mathscr{P}}\boldsymbol{C}} = \boldsymbol{T}_{\mathrm{L}} \boldsymbol{B}^{-1} \mathrm{vec}\{\boldsymbol{C}^{-1}\} = \boldsymbol{T}_{\mathrm{L}} \boldsymbol{\mu}_{\boldsymbol{C}}$$

holds for a certain matrix $\boldsymbol{T}_{\mathrm{L}}$, the proof can be completed by showing that $\|\boldsymbol{T}_{\mathrm{L}}\|_1 \leq 1$, since we have:

$$\|\boldsymbol{T}_{\mathrm{L}}\|_1 = \sup_{\boldsymbol{x}} \frac{\|\boldsymbol{T}_{\mathrm{L}} \boldsymbol{x}\|_1}{\|\boldsymbol{x}\|_1} \geq \frac{\|\mu_{\mathcal{T}_{\mathscr{P}}\boldsymbol{C}}\|_1}{\|\boldsymbol{\mu}_{\boldsymbol{C}}\|_1}.$$

Next we construct the matrix $\boldsymbol{T}_{\mathrm{L}}$ explicitly. Let us consider the QR decomposition [133] of the matrix $\boldsymbol{B}$

$$\boldsymbol{B} = \boldsymbol{Q}\boldsymbol{R}, \tag{A.7}$$

where $\boldsymbol{Q}$ is an orthogonal matrix and $\boldsymbol{R}$ is an upper triangular matrix. Substituting (A.5) and (A.7) into (A.6), we have

$$\mu_{\mathcal{T}_{\mathscr{P}}\boldsymbol{C}} = \frac{1}{4^n} \boldsymbol{I}_{\mathscr{P}} \boldsymbol{R}^{\mathrm{T}} \boldsymbol{Q}^{\mathrm{T}} \mathrm{vec}\{\boldsymbol{C}^{-1}\}. \tag{A.8}$$

Similarly, we can obtain

$$\boldsymbol{\mu}_{\boldsymbol{C}} = \boldsymbol{R}^{-1} \boldsymbol{Q}^{\mathrm{T}} \mathrm{vec}\{\boldsymbol{C}^{-1}\}. \tag{A.9}$$

Having compared (A.8) and (A.9), we may observe that

$$\boldsymbol{T}_{\mathrm{L}} = \frac{1}{4^n} \boldsymbol{I}_{\mathscr{P}} \boldsymbol{R}^{\mathrm{T}} \boldsymbol{R}.$$

Upon introducing $\boldsymbol{Q} = [\boldsymbol{q}_1\ \boldsymbol{q}_2\ \dots\ \boldsymbol{q}_{16^n}]$, the matrix $\boldsymbol{R}$ can be represented by

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{b}_1 & \boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{b}_2 & \boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{b}_3 & \dots \\ 0 & \boldsymbol{q}_2^{\mathrm{T}}\boldsymbol{b}_2 & \boldsymbol{q}_2^{\mathrm{T}}\boldsymbol{b}_3 & \dots \\ 0 & 0 & \boldsymbol{q}_3^{\mathrm{T}}\boldsymbol{b}_3 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{A.10}$$

Since Pauli operators are orthogonal to each other, the first $4^n$ columns in $\boldsymbol{B}$ (i.e., $\boldsymbol{b}_1$ through $\boldsymbol{b}_{4^n}$) are also mutually orthogonal, meaning that

$$\boldsymbol{q}_i^{\mathrm{T}}\boldsymbol{b}_j = \begin{cases} \|\boldsymbol{b}_i\|_2, & i = j; \\ 0, & i \neq j, \end{cases}$$

when $i = 1, 2, \dots, 4^n$ and $j = 1, 2, \dots, 4^n$. Therefore we have

$$\boldsymbol{I}_{\mathscr{P}}\boldsymbol{R}^{\mathrm{T}} = [\mathrm{diag}\,\{\|\boldsymbol{b}_1\|_2, \|\boldsymbol{b}_2\|_2, \dots, \|\boldsymbol{b}_{4^n}\|_2\}\ \boldsymbol{0}_{4^n\times 16^n}].$$

Upon introducing $m_{\boldsymbol{B}} = \max_{i=1,\dots,16^n} \|\boldsymbol{b}_i\|_2$, from (A.10) we have

$$\begin{aligned} \|\boldsymbol{T}_{\mathrm{L}}\|_1 &\leq \frac{m_{\boldsymbol{B}}}{4^n} \max_{i=1,\dots,16^n} \sum_{j=1}^{\min\{4^n,i\}} |\boldsymbol{q}_j^{\mathrm{T}}\boldsymbol{b}_i| && \text{(A.11a)} \\ &\leq \frac{m_{\boldsymbol{B}}}{4^n} \max_{i=1,\dots,16^n} \|\boldsymbol{w}_i\|_1 \max_{j=1,\dots,4^n} \|\boldsymbol{b}_j\|_2 && \text{(A.11b)} \\ &\leq \frac{m_{\boldsymbol{B}}^2}{4^n} \max_{i=1,\dots,16^n} \|\boldsymbol{w}_i\|_1, && \text{(A.11c)} \end{aligned}$$

where $\boldsymbol{w}_i \in \mathbb{R}^{\min\{4^n,i\}}$, and

$$[\boldsymbol{w}_i]_j = \frac{\boldsymbol{b}_j^{\mathrm{T}}\boldsymbol{b}_i}{\|\boldsymbol{b}_j\|_2^2}. \tag{A.12}$$

Using (A.12), the projection of $\boldsymbol{b}_i$ onto the space spanned by Pauli operators can be expressed as

$$\begin{aligned} \boldsymbol{T}_{\mathscr{P}}\boldsymbol{b}_i &= [\boldsymbol{Q}]_{:,1:4^n}[\boldsymbol{Q}]_{:,1:4^n}^{\mathrm{T}}\boldsymbol{b}_i \\ &= \sum_{j=1}^{\min\{4^n,i\}} (\boldsymbol{q}_j^{\mathrm{T}}\boldsymbol{b}_i)\boldsymbol{q}_j \\ &= \sum_{j=1}^{\min\{4^n,i\}} [\boldsymbol{w}_i]_j\boldsymbol{b}_j. \end{aligned}$$

Since $\boldsymbol{B}_i = \mathrm{vec}^{-1}\{\boldsymbol{b}_i\}$ corresponds to a CPTnI operator, the complete positiveness[1] implies $[\boldsymbol{w}_i]_j \geq 0$ for all $i$ and $j$, and the "trace–non-increasing" property implies

---

[1] We say that an operator is complete positive if it maps positive semidefinite matrices to positive semidefinite matrices.

$\sum_j [\boldsymbol{w}_i]_j \leq 1$. Therefore we have $\|\boldsymbol{w}_i\|_1 \leq 1$ for all $i$, and hence

$$\|\boldsymbol{T}_{\mathrm{L}}\|_1 \leq 4^{-n} m_{\boldsymbol{B}}^2. \tag{A.13}$$

Note that $m_{\boldsymbol{B}}^2 = \max_i \|\boldsymbol{B}_i\|_{\mathrm{F}}^2$. By consider the operator-sum decomposition of $\boldsymbol{B}_i$, we see that

$$\begin{aligned}
\|\boldsymbol{B}_i\|_{\mathrm{F}}^2 &= \|\sum_i (\boldsymbol{K}_i^* \otimes \boldsymbol{K}_i)\|_{\mathrm{F}}^2 && \text{(A.14a)}\\
&\leq \left(\sum_i \|\boldsymbol{K}_i^* \otimes \boldsymbol{K}_i\|_{\mathrm{F}}\right)^2 && \text{(A.14b)}\\
&= \left(\sum_i \mathrm{Tr}\{\boldsymbol{K}_i^\dagger \boldsymbol{K}_i\}\right)^2 && \text{(A.14c)}\\
&\leq 4^n, && \text{(A.14d)}
\end{aligned}$$

where (A.14d) follows from (4.10).

Combining (A.13) and (A.14), we can see that $\|\boldsymbol{T}_{\mathrm{L}}\|_1 \leq 1$, hence the proof is completed.

□

## A.2 Proof of Proposition 4.5

To facilitate our further analysis, we denote the Hadamard transform matrix on the space of Pauli transfer matrix representation of an $n$-qubit system as $\boldsymbol{H}_n \in \mathbb{R}^{4^n \times 4^n}$. The corresponding inverse Hadamard transform is denoted by $\boldsymbol{H}_{\mathrm{n}}^{-1} = \frac{1}{4^n}\boldsymbol{H}$. We omit the subscript $n$, whenever there is no confusion. Given these notations, according to (4.19), the simplified quasi-probability representation vector for a Pauli channel $\mathcal{C}$ can be expressed as

$$\tilde{\boldsymbol{\mu}}_{\mathcal{C}} = \boldsymbol{H}^{-1}(1/\boldsymbol{c}). \tag{A.15}$$

Since the channel is CPTP, the vector $1/\boldsymbol{c}$ satisfies $[1/\boldsymbol{c}]_1 = 1$. Hence we define $\boldsymbol{\zeta} \in \mathbb{R}^{4^n - 1}$ so that

$$1/\boldsymbol{c} = \mathbf{1} + [0\ \boldsymbol{\zeta}^{\mathrm{T}}]^{\mathrm{T}}.$$

For comparison, we consider a depolarizing channel $\mathcal{L}$ having quasi-probability representation corresponding to $\tilde{\boldsymbol{\mu}}_{\mathcal{L}}$ given by

$$\tilde{\boldsymbol{\mu}}_{\mathcal{L}} = \boldsymbol{H}^{-1}(\mathbf{1} + [0\ \bar{\zeta}\mathbf{1}^{\mathrm{T}}]^{\mathrm{T}}), \tag{A.16}$$

where $\bar{\zeta} \triangleq \frac{1}{4^n-1}\sum_{i=1}^{4^n-1}\zeta_i$. One may verify that the channel $\mathcal{L}$ characterized by (A.16) is a depolarizing channel with its GGEP satisfying

$$1-\epsilon(\mathcal{L}) = \frac{1}{4^n}\Big(1+\frac{4^n-1}{1+\bar{\zeta}}\Big). \tag{A.17}$$

By contrast, the GGEP of the channel $\mathcal{C}$ satisfies

$$\begin{aligned} 1-\epsilon(\mathcal{C}) &= \frac{1}{4^n}\mathbf{1}^{\mathrm{T}}\boldsymbol{c} \\ &= \frac{1}{4^n}\Big(1+\sum_{i=1}^{4^n-1}\frac{1}{1+\zeta_i}\Big). \end{aligned} \tag{A.18}$$

Since $\zeta_i > -1$, $\forall i$, from (A.17) and (A.18) we have

$$\epsilon(\mathcal{L}) \geq \epsilon(\mathcal{C}) \tag{A.19}$$

due to the convexity of $f(x) = (1+x)^{-1}$, when $x > -1$.

Next we show that $\|\tilde{\boldsymbol{\mu}}_{\mathcal{C}}\|_1 \geq \|\tilde{\boldsymbol{\mu}}_{\mathcal{L}}\|_1$. Note that the vector $\tilde{\boldsymbol{\mu}}_{\mathrm{C}}$ can be decomposed as

$$\tilde{\boldsymbol{\mu}}_{\mathcal{C}} = \tilde{\boldsymbol{\mu}}_{\mathcal{L}} + \boldsymbol{H}^{-1}\boldsymbol{r}, \tag{A.20}$$

where

$$\begin{aligned} \boldsymbol{r} &= 1/\boldsymbol{c} - 1/\boldsymbol{l} \\ &= [0\ (\boldsymbol{\zeta} - \bar{\zeta}\mathbf{1})^{\mathrm{T}}]^{\mathrm{T}}. \end{aligned}$$

From the definition of $\bar{\zeta}$ we see that $\mathbf{1}^{\mathrm{T}}\boldsymbol{r} = 0$, hence $[\tilde{\boldsymbol{\mu}}_{\mathcal{C}}]_1 = [\tilde{\boldsymbol{\mu}}_{\mathcal{L}}]_1 \triangleq \mu_1$. In addition, we have

$$\mathbf{1}^{\mathrm{T}}\tilde{\boldsymbol{\mu}}_{\mathcal{C}} = \mathbf{1}^{\mathrm{T}}\tilde{\boldsymbol{\mu}}_{\mathcal{L}} = [1/\boldsymbol{c}]_1 = 1, \tag{A.21}$$

since the channels are CPTP. Therefore we obtain

$$\mathbf{1}^{\mathrm{T}}[\tilde{\boldsymbol{\mu}}_{\mathcal{C}}]_{2:4^n} = \mathbf{1}^{\mathrm{T}}[\tilde{\boldsymbol{\mu}}_{\mathcal{L}}]_{2:4^n}. \tag{A.22}$$

The 1-norm of $\tilde{\boldsymbol{\mu}}_{\mathcal{L}}$ can be calculated explicitly as

$$\|\tilde{\boldsymbol{\mu}}_{\mathcal{L}}\|_1 = \mu_1 - \sum_{i=2}^{4^n}[\tilde{\boldsymbol{\mu}}_{\mathcal{L}}]_i. \tag{A.23}$$

For $\tilde{\boldsymbol{\mu}}_{\mathcal{C}}$, we denote the sign of its $i$-th entry as $s_i$, thus

$$\begin{aligned} \|\tilde{\boldsymbol{\mu}}_{\mathcal{C}}\|_1 &= \mu_1 + \sum_{i=2}^{4^n} s_i[\tilde{\boldsymbol{\mu}}_{\mathcal{C}}]_i \\ &= \mu_1 + \mathbf{1}_+^{\mathrm{T}}[\tilde{\boldsymbol{\mu}}_{\mathcal{C}}]_{2:4^n} - \mathbf{1}_-^{\mathrm{T}}[\tilde{\boldsymbol{\mu}}_{\mathcal{C}}]_{2:4^n}, \end{aligned} \tag{A.24}$$

where

$$[\mathbf{1}_{+}]_i = \begin{cases} 1, & s_{i-1} > 0; \\ 0, & s_{i-1} < 0. \end{cases} \text{ and } [\mathbf{1}_{-}]_i = \begin{cases} 1, & s_{i-1} < 0; \\ 0, & s_{i-1} > 0. \end{cases}$$

From (A.22) we have

$$\mathbf{1}_{+}^{\mathrm{T}}[\tilde{\boldsymbol{\mu}}_{\mathcal{C}}]_{2:4^n} + \mathbf{1}_{-}^{\mathrm{T}}[\tilde{\boldsymbol{\mu}}_{\mathcal{C}}]_{2:4^n} = \sum_{i=2}^{4^n}[\tilde{\boldsymbol{\mu}}_{\mathcal{L}}]_i.$$

Hence

$$\begin{aligned} \|\tilde{\boldsymbol{\mu}}_{\mathcal{C}}\|_1 &= \|\tilde{\boldsymbol{\mu}}_{\mathcal{L}}\|_1 + 2\mathbf{1}_{+}^{\mathrm{T}}[\tilde{\boldsymbol{\mu}}_{\mathcal{C}}]_{2:4^n} \\ &\geq \|\tilde{\boldsymbol{\mu}}_{\mathcal{L}}\|_1. \end{aligned} \tag{A.25}$$

Finally, since $\epsilon(\mathcal{L}) \geq \epsilon(\mathcal{C})$, we may construct a depolarizing channel $\mathcal{L}'$ characterized by $\epsilon(\mathcal{L}') = \epsilon(\mathcal{C})$, while satisfying

$$\|\tilde{\boldsymbol{\mu}}_{\mathcal{L}}\|_1 \geq \|\tilde{\boldsymbol{\mu}}_{\mathcal{L}'}\|_1.$$

Hence the proof is completed.

## A.3 Proof of Proposition 4.7

*Proof.* From (4.28), we have

$$\gamma_{\mathcal{C}} = \|\boldsymbol{C}^{-1}\boldsymbol{\alpha}\|_1^2 - 1, \tag{A.26}$$

where

$$\boldsymbol{C} = (1-\epsilon)\boldsymbol{I} + \boldsymbol{A}. \tag{A.27}$$

Since the graph $\mathscr{G}$ is symmetric, each column (resp. row) of $\boldsymbol{C}$ can be obtained by permuting the first column (resp. row) of $\boldsymbol{C}$. Thus we have

$$\gamma_{\mathcal{C}} = \|\boldsymbol{C}^{-1}\|_1^2 - 1. \tag{A.28}$$

From (A.27) we can obtain

$$\begin{aligned} \boldsymbol{C}^{-1} &= ((1-\epsilon)\boldsymbol{I} + \epsilon\tilde{\boldsymbol{A}})^{-1} && \text{(A.29a)} \\ &= \frac{1}{1-\epsilon}\left(\boldsymbol{I} + \frac{\epsilon}{1-\epsilon}\tilde{\boldsymbol{A}}\right)^{-1} && \text{(A.29b)} \\ &= \frac{1}{1-\epsilon}\sum_{n=0}^{\infty}(-1)^n\left(\frac{\epsilon}{1-\epsilon}\right)^n\tilde{\boldsymbol{A}}^n, && \text{(A.29c)} \end{aligned}$$

where $\tilde{\boldsymbol{A}} \triangleq \epsilon^{-1}\boldsymbol{A}$, and (A.29c) is obtained using the matrix inversion lemma. Exploiting the sub-multiplicativity of matrix $p$-norms [214, Chap. 5], we have

$$\|\tilde{\boldsymbol{A}}^n\|_1 \leq \|\tilde{\boldsymbol{A}}\|_1^n = 1, \tag{A.30}$$

where the equality follows from the fact that $\mathbf{1}^{\mathrm{T}}\tilde{\boldsymbol{A}} = 1$ and that all entries in $\tilde{\boldsymbol{A}}$ are non-negative. Substituting (A.30) into (A.29), we have

$$\|\boldsymbol{C}^{-1}\|_1 \leq \frac{1}{1-\epsilon}\sum_{n=0}^{\infty}\left\|(-1)^n\left(\frac{\epsilon}{1-\epsilon}\right)^n\tilde{\boldsymbol{A}}^n\right\|_1 \tag{A.31a}$$

$$\leq \frac{1}{1-\epsilon}\sum_{n=0}^{\infty}\left(\frac{\epsilon}{1-\epsilon}\right)^n = \frac{1}{1-2\epsilon}. \tag{A.31b}$$

Therefore, from (A.28) we obtain

$$\gamma_{\mathcal{C}} \leq \left(\frac{1}{1-2\epsilon}\right)^2 - 1 = 4\epsilon\cdot\frac{1-\epsilon}{(1-2\epsilon)^2}. \tag{A.32}$$

To show that channels having a single type of error achieve the equality, we note that

$$\mathcal{P}_i^{2n} = \mathcal{I},\ \mathcal{P}_i^{2n+1} = \mathcal{P}_i \tag{A.33}$$

holds for any Pauli operator $\mathcal{P}_i$. In light of this, the inverse of their PRW matrix can be shown to satisfy

$$\boldsymbol{C}^{-1} = \frac{1}{1-\epsilon}\sum_{n=0}^{\infty}\left\{\left(\frac{\epsilon}{1-\epsilon}\right)^{2n}\boldsymbol{I} - \left(\frac{\epsilon}{1-\epsilon}\right)^{2n+1}\tilde{\boldsymbol{A}}\right\}.$$

Therefore we have

$$\begin{aligned}\|\boldsymbol{C}^{-1}\|_1 &= \frac{1}{1-\epsilon}\sum_{n=0}^{\infty}\left\{\left(\frac{\epsilon}{1-\epsilon}\right)^{2n} + \left(\frac{\epsilon}{1-\epsilon}\right)^{2n+1}\|\tilde{\boldsymbol{A}}\|_1\right\}\\ &= \frac{1}{1-\epsilon}\sum_{n=0}^{\infty}\left(\frac{\epsilon}{1-\epsilon}\right)^n,\end{aligned}$$

which is identical to (A.31b). Hence the proof is completed. □

## A.4 Proof of Proposition 4.8

*Proof.* Let $\boldsymbol{\eta}$ be the probability vector of a Pauli channel. We first show that the function

$$f(\boldsymbol{\eta}) = \left\|\boldsymbol{H}_1^{-1}\left(1/(\boldsymbol{H}_1[1-\varepsilon\,\boldsymbol{\eta}^{\mathrm{T}}]^{\mathrm{T}})\right)\right\|_1 \tag{A.34}$$

is Schur-convex with respect to $\boldsymbol{\eta}$. We proceed by first decomposing $f(\boldsymbol{\eta})$ as

$$f(\boldsymbol{\eta}) = g\{h_1[h_2(\boldsymbol{\eta})]\}, \tag{A.35}$$

where

$$\begin{aligned} g(\boldsymbol{x}) &= \left\| \boldsymbol{H}_1^{-1} [1\ \boldsymbol{x}^{\mathrm{T}}]^{\mathrm{T}} \right\|_1 \quad (\boldsymbol{x} \succcurlyeq \mathbf{1}), && \text{(A.36a)} \\ h_1(\boldsymbol{x}) &= 1/\boldsymbol{x}\ \ (\mathbf{0} \preccurlyeq \boldsymbol{x} \preccurlyeq \mathbf{1}), && \text{(A.36b)} \\ h_2(\boldsymbol{\eta}) &= \boldsymbol{H}_1 [1 - \epsilon\ \boldsymbol{\eta}^{\mathrm{T}}]^{\mathrm{T}} \ \ \left( \mathbf{0} \preccurlyeq \boldsymbol{\eta} \preccurlyeq \frac{\epsilon}{3} \mathbf{1} \right). && \text{(A.36c)} \end{aligned}$$

Since $h_2(\boldsymbol{\eta})$ is an affine function of $\boldsymbol{\eta}$, we see that

$$\begin{aligned} h(\boldsymbol{\eta}) &= h_1[h_2(\boldsymbol{\eta})] \\ &= 1/\left( \boldsymbol{H}_1 [1 - \epsilon\ \boldsymbol{\eta}^{\mathrm{T}}]^{\mathrm{T}} \right) \end{aligned} \tag{A.37}$$

is element-wise convex with respect to $\boldsymbol{\eta}$. Therefore, to show that $f(\boldsymbol{\eta})$ is Schur-convex, it suffices to show that $g(\boldsymbol{x})$ is Schur-convex and increasing for $\boldsymbol{x}$ satisfying $\boldsymbol{x} \succcurlyeq \mathbf{1}$.

Next we show the Schur-convexity of $g(\boldsymbol{x})$. Note that

$$\begin{aligned} g(\boldsymbol{x}) &= \left\| \boldsymbol{H}_1^{-1} [1\ \boldsymbol{x}^{\mathrm{T}}]^{\mathrm{T}} \right\|_1 && \text{(A.38a)} \\ &= 1 + \mathbf{1}^{\mathrm{T}} (\boldsymbol{x} - 1) + \frac{1}{4} \left\| \tilde{\boldsymbol{H}}_1 (\boldsymbol{x} - 1) \right\|_1, && \text{(A.38b)} \end{aligned}$$

where

$$\tilde{\boldsymbol{H}}_1 = \begin{bmatrix} -1 & 1 & -1 \\ 1 & -1 & -1 \\ -1 & -1 & 1 \end{bmatrix} \tag{A.39}$$

is obtained by removing the first row and the first column from $\boldsymbol{H}_1$. Since doubly stochastic transformations do not affect the term $\mathbf{1}^{\mathrm{T}}(\boldsymbol{x} - 1)$, the problem is reduced to showing the Schur-convexity of $\|\tilde{\boldsymbol{H}}_1 \boldsymbol{x}\|_1$ for $\boldsymbol{x} \succcurlyeq \mathbf{0}$. To facilitate the analysis, we utilize $\epsilon = \mathbf{1}^{\mathrm{T}} \boldsymbol{x}$ and define $\boldsymbol{x} = [x_1\ x_2\ x_3]^{\mathrm{T}}$. Now we see that

$$\tilde{\boldsymbol{H}}_1 \boldsymbol{x} = [2x_2 - \epsilon\ \ 2x_1 - \epsilon\ \ 2x_3 - \epsilon]^{\mathrm{T}}, \tag{A.40}$$

hence

$$\|\tilde{\boldsymbol{H}}_1 \boldsymbol{x}\|_1 = \|2\boldsymbol{x} - \epsilon\|_1. \tag{A.41}$$

For fixed $\epsilon$, $\|2\boldsymbol{x} - \epsilon\|_1$ is convex with respect to $\boldsymbol{x}$. In addition, it is also a symmetric function of $\boldsymbol{x}$, meaning that its value is unchanged upon permutation of $\boldsymbol{x}$. Therefore, $g(\boldsymbol{x})$ is Schur-convex.

To show that $g(\boldsymbol{x})$ is increasing, we calculate the gradient of $g(\boldsymbol{x})$ as

$$\begin{aligned} \nabla_{\boldsymbol{x}} g(\boldsymbol{x}) &= \frac{\partial}{\partial \boldsymbol{x}} \left( \mathbf{1}^{\mathrm{T}} (\boldsymbol{x} - 1) + \|\tilde{\boldsymbol{H}}_1 (\boldsymbol{x} - 1)\|_1 \right) \\ &= \mathbf{1} + \tilde{\boldsymbol{H}}_1 \mathrm{sgn}(\tilde{\boldsymbol{H}}_1 (\boldsymbol{x} - 1)) \end{aligned} \tag{A.42}$$

where $\mathrm{sgn}(\cdot)$ is the sign function satisfying

$$[\mathrm{sgn}(\boldsymbol{x})]_i = \begin{cases} -1, & x_i < 0; \\ 1, & x_i > 0; \\ 0, & x_i = 0. \end{cases}$$

After some manipulation, one can verify that $\tilde{\boldsymbol{H}}_1 \mathrm{sgn}(\tilde{\boldsymbol{H}}_1(\boldsymbol{x} - 1)) \succcurlyeq -\mathbf{1}$ according to (A.39), hence $g(\boldsymbol{x})$ is increasing.

Given that the SOF of single-qubit Pauli channels is Schur-convex, we may generalize the result to $n$-qubit memoryless Pauli channels. To elaborate, the PRW matrix of an $n$-qubit memoryless Pauli channel can be expressed as

$$\boldsymbol{C}(\boldsymbol{\eta}_{\mathcal{C}}) = \bigotimes_{i=1}^{n} \boldsymbol{C}_i(\boldsymbol{\eta}_{\mathcal{C}_i}), \tag{A.43}$$

where $\boldsymbol{C}_i(\boldsymbol{\eta}_{\mathcal{C}_i})$ corresponds to the partial channel of the $i$-th qubit. Thus

$$\left\| \boldsymbol{C}^{-1}(\boldsymbol{\eta}_{\mathcal{C}}) \right\|_1 = \prod_{i=1}^{n} \| \boldsymbol{C}_i^{-1}(\boldsymbol{\eta}_{\mathcal{C}_i}) \|_1. \tag{A.44}$$

Since $\| \boldsymbol{C}_i^{-1}(\boldsymbol{\eta}_{\mathcal{C}_i}) \|_1$ is Schur-convex with respect to the corresponding probability vector $\boldsymbol{\eta}_{\mathcal{C}_i}$, we have

$$\| \boldsymbol{C}_i^{-1}(\boldsymbol{\eta}_{\mathcal{C}_i}) \|_1 \geq \| \boldsymbol{C}_i^{-1}(\boldsymbol{Q}_i \boldsymbol{\eta}_{\mathcal{C}_i}) \|_1 \tag{A.45}$$

for any doubly stochastic matrix $\boldsymbol{Q}_i$. Therefore

$$\left\| \boldsymbol{C}^{-1}(\boldsymbol{\eta}_{\mathcal{C}}) \right\|_1 \geq \prod_{i=1}^{n} \left\| \boldsymbol{C}_i^{-1}(\boldsymbol{Q}_i \boldsymbol{\eta}_{\mathcal{C}_i}) \right\|_1. \tag{A.46}$$

Hence the proof is completed. □

## A.5 Proof of Proposition 5.1

*Proof.* First observe that the matrix representation of a perfect gate $\boldsymbol{G}_i$ as well as that of a channel $\boldsymbol{C}_i$ take the following block-diagonal form

$$\boldsymbol{G}_i = \begin{pmatrix} 1 & \mathbf{0}^{\mathrm{T}} \\ \mathbf{0} & \boldsymbol{U}_i \end{pmatrix}, \quad \boldsymbol{C}_i = \begin{pmatrix} 1 & \mathbf{0}^{\mathrm{T}} \\ \mathbf{0} & \boldsymbol{D}_i \end{pmatrix}, \tag{A.47}$$

where $\boldsymbol{U}_i$ is a unitary matrix, whereas $\boldsymbol{D}_i$ is a diagonal matrix having diagonal entries taking values in the interval $[0, 1]$. Since the matrix $\boldsymbol{R}_{N_{\mathrm{G}}}$ is the product of several $\boldsymbol{G}_i$ and $\boldsymbol{C}_i$, it becomes clear that its largest singular values satisfies $\sigma_1(\boldsymbol{R}_{N_{\mathrm{G}}}) = 1$, and its second largest singular value satisfies

$$\sigma_2(\boldsymbol{R}_{N_\mathrm{G}}) \leq \prod_{i=1}^{N_\mathrm{G}} \|\boldsymbol{D}_i\|_2. \tag{A.48}$$

Furthermore, we have

$$\left| r - \frac{1}{2^n}\mathrm{Tr}\left\{\mathcal{M}_\mathrm{ob}\right\} \right| \leq \sigma_2(\boldsymbol{R}_{N_\mathrm{G}}) \tag{A.49}$$

due to the "bounded observable" Assumption 3.

Note that the quantity $N_\mathrm{L}$ defined in this proposition is related to the depth of the circuit. To elaborate, if we say that "a layer of gates" is executed if each qubit has been processed by at least one gate, then the entire circuit consists of at least $N_\mathrm{L}$ layers. For each single-qubit channel $\boldsymbol{C}$ in these layers, due to the assumption that each single-qubit Pauli error occurs at probability of at least $\epsilon_1$, the following bound holds:

$$\begin{aligned} \boldsymbol{C} &= \boldsymbol{I} - 2\mathrm{diag}\left\{[p_\mathrm{X} + p_\mathrm{Z}\ p_\mathrm{Y} + p_\mathrm{Z}\ p_\mathrm{X} + p_\mathrm{Y}]\right\} \\ &\preccurlyeq (1 - 4\epsilon_1), \end{aligned} \tag{A.50}$$

where $p_\mathrm{X}$, $p_\mathrm{Y}$ and $p_\mathrm{Z}$ are error probabilities corresponding to the Pauli-X, Y and Z errors, respectively. Thus we obtain

$$\begin{aligned} \sigma_2(\boldsymbol{R}_{N_\mathrm{G}}) &\leq (1 - 4\epsilon_1)^{N_\mathrm{L}} \\ &= \exp\{N_\mathrm{L}\ln(1 - 4\epsilon_1)\} \\ &\leq \exp(-4\epsilon_1 N_\mathrm{L}). \end{aligned} \tag{A.51}$$

Hence the proof is completed. □

## A.6 Proof of Proposition 5.2

*Proof.* In this proof, we will work under the operator-sum representation of quantum channels. Since we consider Pauli channels, the recursion (5.3) may be rewritten as

$$\begin{aligned} \rho_k &= \sum_{i=1}^{4^n} [\boldsymbol{p}_k]_i \mathcal{S}_i \mathcal{G}_k \rho_{k-1} \mathcal{G}_k^\dagger \mathcal{S}_i \\ &= [\boldsymbol{p}_k]_1 \mathcal{G}_k \rho_{k-1} \mathcal{G}_k^\dagger + \sum_{i=2}^{4^n} [\boldsymbol{p}_k]_i \mathcal{S}_i \mathcal{G}_k \rho_{k-1} \mathcal{G}_k^\dagger \mathcal{S}_i. \end{aligned} \tag{A.52}$$

Assumption 3 implies that $\|\mathcal{M}_\mathrm{ob}\|_2 \leq 1$, meaning that

$$\mathrm{Tr}\left\{\mathcal{M}_\mathrm{ob}\rho\right\} \leq 1$$

holds for any legitimate density matrix $\rho$. Note that terms such as $\mathcal{S}_i\mathcal{G}_k\rho_{k-1}\mathcal{G}_k^\dagger\mathcal{S}_i$ in (A.52) are indeed legitimate density matrices. Thus the computational result satisfies

$$\begin{aligned}|\widetilde{r} - \mathrm{Tr}\left\{\mathcal{M}_{\mathrm{ob}}\rho_{N_{\mathrm{G}}}\right\}| &\leq \left(|\widetilde{r}| + \|\mathcal{M}_{\mathrm{ob}}\|_2\right)\left(1 - \prod_{k=1}^{N_{\mathrm{G}}}[\boldsymbol{p}_k]_1\right)\\ &\leq 2\left(1 - \prod_{k=1}^{N_{\mathrm{G}}}[\boldsymbol{p}_k]_1\right).\end{aligned} \tag{A.53}$$

According to Assumption 2, for any $k$, the vector $\boldsymbol{p}_k$ satisfies

$$\begin{aligned}&[\boldsymbol{p}_k]_1 \geq 1 - \epsilon_{\mathrm{u}},\\ &\sum_{i=2}^{4^n}[\boldsymbol{p}_k]_i \leq \epsilon_{\mathrm{u}}.\end{aligned} \tag{A.54}$$

Therefore, we have

$$\begin{aligned}|\widetilde{r} - \mathrm{Tr}\left\{\mathcal{M}_{\mathrm{ob}}\rho_{N_{\mathrm{G}}}\right\}| &\leq 2(1 - (1 - \epsilon_{\mathrm{u}})^{N_{\mathrm{G}}})\\ &\leq 2\epsilon_{\mathrm{u}}N_{\mathrm{G}}.\end{aligned} \tag{A.55}$$

Hence the proof is completed. □

## A.7 Proof of Proposition 5.3

*Proof.* We first expand the expression of MSE as follows

$$\begin{aligned}\mathbb{E}\{(\mathsf{r} - \widetilde{r})^2\} &= \mathbb{E}\left\{\left(\boldsymbol{v}_{\mathrm{ob}}^{\mathrm{T}}\mathbf{v}_{N_{\mathrm{G}}} - \widetilde{r}\right)^2\right\}\\ &= \boldsymbol{v}_{\mathrm{ob}}^{\mathrm{T}}\mathbb{E}\left\{\mathbf{v}_{N_{\mathrm{G}}}\mathbf{v}_{N_{\mathrm{G}}}^{\mathrm{T}}\right\}\boldsymbol{v}_{\mathrm{ob}} + \widetilde{r}^2 - 2\widetilde{r}\boldsymbol{v}_{\mathrm{ob}}^{\mathrm{T}}\mathbb{E}\{\mathbf{v}_{N_{\mathrm{G}}}\}.\end{aligned} \tag{A.56}$$

Hence the RMSE is given by

$$\sqrt{\mathbb{E}\{(\mathsf{r} - \widetilde{r})^2\}} = \sqrt{\boldsymbol{v}_{\mathrm{ob}}^{\mathrm{T}}\boldsymbol{A}_k\boldsymbol{v}_{\mathrm{ob}} + \widetilde{r}^2 - 2\widetilde{r}\boldsymbol{v}_{\mathrm{ob}}^{\mathrm{T}}\boldsymbol{\mu}_k}, \tag{A.57}$$

where $\boldsymbol{A}_k := \mathbb{E}\{\mathbf{v}_k\mathbf{v}_k^{\mathrm{T}}\}$ and $\boldsymbol{\mu}_k := \mathbb{E}\{\mathbf{v}_k\}$.

Using (5.40), we have

$$\mathbf{v}_k = \widetilde{\mathbf{c}}_k \odot (\boldsymbol{G}_k\mathbf{v}_{k-1}). \tag{A.58}$$

This implies the following recursive relationships:

$$\begin{aligned}\boldsymbol{A}_k &= \mathbb{E}\{\widetilde{\mathbf{c}}_k\widetilde{\mathbf{c}}_k^{\mathrm{T}}\} \odot \boldsymbol{G}_k\boldsymbol{A}_{k-1}\boldsymbol{G}_k^{\mathrm{T}},\\ \boldsymbol{\mu}_k &= \boldsymbol{G}_k\boldsymbol{\mu}_{k-1}.\end{aligned} \tag{A.59}$$

In fact, we have

$$\mathbb{E}\{\widetilde{\mathbf{c}}_k\widetilde{\mathbf{c}}_k^{\mathrm{T}}\} = \mathbf{1}\mathbf{1}^{\mathrm{T}} + \boldsymbol{\Xi}_k, \tag{A.60}$$

which follows from (5.38). Substituting (A.60) into (A.59), we obtain

$$\boldsymbol{A}_k = \boldsymbol{G}_k\boldsymbol{A}_{k-1}\boldsymbol{G}_k^{\mathrm{T}} + \boldsymbol{\Xi}_k \odot \boldsymbol{G}_k\boldsymbol{A}_{k-1}\boldsymbol{G}_k^{\mathrm{T}}. \tag{A.61}$$

Let us now consider the case of $k = 1$. In this case, $v_0$ of (5.8) is a deterministic vector, thus we have

$$\begin{aligned} \boldsymbol{A}_0 &= \boldsymbol{v}_0\boldsymbol{v}_0^{\mathrm{T}}, \\ \boldsymbol{\mu}_0 &= \boldsymbol{v}_0. \end{aligned} \tag{A.62}$$

Using the recursive relationship of $\boldsymbol{\mu}_k = \boldsymbol{G}_k\boldsymbol{\mu}_{k-1}$, we now see that $\boldsymbol{v}_{\mathrm{ob}}^{\mathrm{T}}\boldsymbol{\mu}_{N_{\mathrm{G}}} = \widetilde{r}$. Hence we may simplify (A.56) as

$$\begin{aligned} \mathbb{E}\{(\mathsf{r}-\widetilde{r})^2\} &= \boldsymbol{v}_{\mathrm{ob}}^{\mathrm{T}}\boldsymbol{A}_{N_{\mathrm{G}}}\boldsymbol{v}_{\mathrm{ob}} + \widetilde{r}^2 - 2\widetilde{r}\boldsymbol{v}_{\mathrm{ob}}^{\mathrm{T}}\boldsymbol{\mu}_{N_{\mathrm{G}}} \\ &= \boldsymbol{v}_{\mathrm{ob}}^{\mathrm{T}}\boldsymbol{A}_{N_{\mathrm{G}}}\boldsymbol{v}_{\mathrm{ob}} - (\boldsymbol{v}_{\mathrm{ob}}^{\mathrm{T}}\boldsymbol{\mu}_{N_{\mathrm{G}}})^2 \\ &= \boldsymbol{v}_{\mathrm{ob}}^{\mathrm{T}}\left(\boldsymbol{A}_{N_{\mathrm{G}}} - \boldsymbol{\mu}_{N_{\mathrm{G}}}\boldsymbol{\mu}_{N_{\mathrm{G}}}^{\mathrm{T}}\right)\boldsymbol{v}_{\mathrm{ob}}. \end{aligned} \tag{A.63}$$

Observe that the term $\boldsymbol{A}_{N_{\mathrm{G}}} - \boldsymbol{\mu}_{N_{\mathrm{G}}}\boldsymbol{\mu}_{N_{\mathrm{G}}}^{\mathrm{T}}$ is in fact the covariance matrix of $\mathbf{v}_{N_{\mathrm{G}}}$, upon defining

$$\boldsymbol{\Sigma}_k := \boldsymbol{A}_k - \boldsymbol{\mu}_k\boldsymbol{\mu}_k^{\mathrm{T}}, \tag{A.64}$$

and substituting into (A.63) we arrive at

$$\mathbb{E}\{(\mathsf{r}-\widetilde{r})^2\} = \boldsymbol{v}_{\mathrm{ob}}^{\mathrm{T}}\boldsymbol{\Sigma}_{N_{\mathrm{G}}}\boldsymbol{v}_{\mathrm{ob}}. \tag{A.65}$$

The covariance matrix can be further formulated as

$$\begin{aligned} \boldsymbol{\Sigma}_k &= \boldsymbol{A}_k - \boldsymbol{\mu}_k\boldsymbol{\mu}_k^{\mathrm{T}} \\ &= \boldsymbol{A}_k - \widetilde{\boldsymbol{R}}_k\boldsymbol{v}_0\boldsymbol{v}_0^{\mathrm{T}}\widetilde{\boldsymbol{R}}_k^{\mathrm{T}}. \end{aligned} \tag{A.66}$$

It now suffices to compute $\boldsymbol{A}_k$. Taking trace from both sides of (A.61), we have

$$\begin{aligned} \mathrm{Tr}\{\boldsymbol{A}_k\} &= \mathrm{Tr}\left\{\boldsymbol{G}_k\boldsymbol{A}_{k-1}\boldsymbol{G}_k^{\mathrm{T}} + \boldsymbol{\Xi}_k \odot \boldsymbol{G}_k\boldsymbol{A}_{k-1}\boldsymbol{G}_k^{\mathrm{T}}\right\} \\ &= \mathrm{Tr}\{\boldsymbol{A}_{k-1}\} + \mathrm{Tr}\left\{\boldsymbol{\Xi}_k \odot \boldsymbol{G}_k\boldsymbol{A}_{k-1}\boldsymbol{G}_k^{\mathrm{T}}\right\} \\ &\leq \mathrm{Tr}\{\boldsymbol{A}_{k-1}\}\,(1 + \|\boldsymbol{\Xi}_k\|_{\max}), \end{aligned} \tag{A.67}$$

where the second line follows from the fact that unitary transformations preserve the trace, and $\|\cdot\|_{\max}$ is defined as

$$\|\boldsymbol{A}\|_{\max} := \max_{i,j} |[\boldsymbol{A}]_{i,j}|.$$

From (5.38) we obtain

$$\begin{aligned}\|\boldsymbol{\Xi}_k\|_{\max} &= \frac{1}{N_{\mathrm{s}}}\|\widetilde{\boldsymbol{H}}\left(\boldsymbol{P}_k - \boldsymbol{p}_k\boldsymbol{p}_k^{\mathrm{T}}\right)\widetilde{\boldsymbol{H}} \odot \boldsymbol{c}_k\boldsymbol{c}_k^{\mathrm{T}}\|_{\max} \\ &\leq \frac{1}{N_{\mathrm{s}}}\|\widetilde{\boldsymbol{H}}\left(\boldsymbol{P}_k - \boldsymbol{p}_k\boldsymbol{p}_k^{\mathrm{T}}\right)\widetilde{\boldsymbol{H}}\|_{\max}\|\boldsymbol{c}_k\boldsymbol{c}_k^{\mathrm{T}}\|_{\max} \\ &\leq \frac{1}{N_{\mathrm{s}}}\left(\|\widetilde{\boldsymbol{H}}\boldsymbol{P}_k\widetilde{\boldsymbol{H}}\|_{\max} + \|\widetilde{\boldsymbol{H}}\boldsymbol{p}_k\boldsymbol{p}_k^{\mathrm{T}}\widetilde{\boldsymbol{H}}\|_{\max}\right),\end{aligned} \tag{A.68}$$

where the third line follows from the fact that $\boldsymbol{c}_k$ represents a contractive transformation, so that $\boldsymbol{c}_k \preccurlyeq \mathbf{1}$. Note that every entry in $\widetilde{\boldsymbol{H}}$ has an absolute value of 1, and hence

$$\|\widetilde{\boldsymbol{H}}\boldsymbol{P}_k\widetilde{\boldsymbol{H}}\|_{\max} \leq \|\mathrm{vec}\{\boldsymbol{P}_k\}\|_1 = 1, \tag{A.69}$$

and

$$\|\widetilde{\boldsymbol{H}}\boldsymbol{p}_k\boldsymbol{p}_k^{\mathrm{T}}\widetilde{\boldsymbol{H}}\|_{\max} \leq \|\mathrm{vec}\{\boldsymbol{p}_k\boldsymbol{p}_k^{\mathrm{T}}\}\|_1 = 1. \tag{A.70}$$

In light of this, the upper bound of $\mathrm{Tr}\{\boldsymbol{A}_k\}$ can now be simplified as follows:

$$\mathrm{Tr}\{\boldsymbol{A}_k\} \leq \mathrm{Tr}\{\boldsymbol{A}_{k-1}\}\left(1 + \frac{2}{N_{\mathrm{s}}}\right). \tag{A.71}$$

From (A.62) we have $\mathrm{Tr}\{\boldsymbol{A}_0\} = 1$ since $\boldsymbol{v}_0$ is a unit vector, hence we obtain

$$\begin{aligned}\mathrm{Tr}\{\boldsymbol{A}_{N_{\mathrm{G}}}\} &\leq \prod_{k=1}^{N_{\mathrm{G}}}\left(1 + \frac{2}{N_{\mathrm{s}}}\right) \\ &\leq \exp\left(2N_{\mathrm{G}}N_{\mathrm{s}}^{-1}\right).\end{aligned} \tag{A.72}$$

Using (A.66), we have

$$\begin{aligned}\mathrm{Tr}\{\boldsymbol{\Sigma}_{N_{\mathrm{G}}}\} &= \mathrm{Tr}\{\boldsymbol{A}_{N_{\mathrm{G}}}\} - \mathrm{Tr}\left\{\boldsymbol{v}_0\boldsymbol{v}_0^{\mathrm{T}}\right\} \\ &\leq \exp\left(2N_{\mathrm{G}}N_{\mathrm{s}}^{-1}\right) - 1.\end{aligned} \tag{A.73}$$

Note that $\boldsymbol{\Sigma}_{\boldsymbol{N}_{\mathrm{G}}}$ is a positive semidefinite matrix, hence we have

$$\mathrm{Tr}\{\boldsymbol{\Sigma}_{\boldsymbol{N}_{\mathrm{G}}}\} \geq \lambda_{\max}(\boldsymbol{\Sigma}_{\boldsymbol{N}_{\mathrm{G}}}), \tag{A.74}$$

where $\lambda_{\max}(\cdot)$ denotes the maximum eigenvalue of a matrix. This implies that

$$\begin{aligned}\sqrt{\mathbb{E}\{(\mathsf{r} - \widetilde{r})^2\}} &= \sqrt{\boldsymbol{v}_{\mathrm{ob}}^{\mathrm{T}}\boldsymbol{\Sigma}_{N_{\mathrm{G}}}\boldsymbol{v}_{\mathrm{ob}}} \\ &\leq \sqrt{\mathrm{Tr}\{\boldsymbol{\Sigma}_{\boldsymbol{N}_{\mathrm{G}}}\}} \cdot \|\boldsymbol{v}_{\mathrm{ob}}\| \\ &\leq \sqrt{\exp\left(2N_{\mathrm{G}}N_{\mathrm{s}}^{-1}\right) - 1} \cdot \|\boldsymbol{v}_{\mathrm{ob}}\|.\end{aligned} \tag{A.75}$$

Hence the proof is completed by applying (5.23). □

## A.8 Proof of Proposition 5.4

*Proof.* We start the proof from revisiting the inequality in (A.68), and arrive at:

$$\begin{aligned}\|\Xi_k\|_{\max} &\le \frac{1}{N_{\mathrm{s}}}\|\widetilde{\boldsymbol{H}}\left(\boldsymbol{P}_k - \boldsymbol{p}_k\boldsymbol{p}_k^{\mathrm{T}}\right)\widetilde{\boldsymbol{H}}\|_{\max} \\ &\le \frac{1}{N_{\mathrm{s}}}\|\mathrm{vec}\{\boldsymbol{P}_k - \boldsymbol{p}_k\boldsymbol{p}_k^{\mathrm{T}}\}\|_1.\end{aligned} \tag{A.76}$$

Next we construct an upper bound for the term $\|\mathrm{vec}\{\boldsymbol{P}_k - \boldsymbol{p}_k\boldsymbol{p}_k^{\mathrm{T}}\}\|_1$. According to the sampling overhead of QEM in [1], Assumption 2 implies that

$$\|\boldsymbol{\alpha}\|_1 \le \sqrt{1+\sigma_{\mathrm{u}}}. \tag{A.77}$$

Since $\alpha_k^{(1)} \ge 1$, we have

$$\sum_{i\neq 1}|\alpha_k^{(i)}| \le \sqrt{1+\sigma_{\mathrm{u}}} - 1. \tag{A.78}$$

This further implies that

$$p_k^{(1)} \ge \frac{1}{\sqrt{1+\sigma_{\mathrm{u}}}}, \quad \sum_{i\neq 1} p_k^{(i)} \le \sqrt{1+\sigma_{\mathrm{u}}} - 1. \tag{A.79}$$

Therefore, upon taking the entry-wise absolute value, we obtain

$$\left|\boldsymbol{P}_k - \boldsymbol{p}_k\boldsymbol{p}_k^{\mathrm{T}}\right| \le \begin{pmatrix} \frac{\sigma_{\mathrm{u}}}{1+\sigma_{\mathrm{u}}} & p_k^{(2)} & \cdots & p_k^{(4^n)} \\ p_k^{(2)} & p_k^{(2)} & p_k^{(2)}p_k^{(3)} & \cdots \\ \vdots & p_k^{(3)}p_k^{(2)} & \ddots & \vdots \\ p_k^{(4^n)} & \vdots & \cdots & p_k^{(4^n)} \end{pmatrix}. \tag{A.80}$$

Here, the symbol "$\le$" stands for entry-wise "not larger than". Observe that summing up the first row, the first column and the main diagonal, by applying (A.79), we see that

$$\|\mathrm{vec}\{\boldsymbol{P}_k - \boldsymbol{p}_k\boldsymbol{p}_k^{\mathrm{T}}\}\|_1 \le 3\left(\sqrt{1+\sigma_{\mathrm{u}}} - 1\right) + \frac{\sigma_{\mathrm{u}}}{1+\sigma_{\mathrm{u}}} + \|\mathrm{vec}\{\boldsymbol{q}_k\boldsymbol{q}_k^{\mathrm{T}}\}\|_1, \tag{A.81}$$

where $\boldsymbol{q}_k := [p_k^{(2)} \ \ldots \ p_k^{(4^q)}]^{\mathrm{T}} \in \mathbb{R}^{4^q-1}$. Note that

$$\begin{aligned}\|\mathrm{vec}\{\boldsymbol{q}_k\boldsymbol{q}_k^{\mathrm{T}}\}\|_1 &= \mathbf{1}^{\mathrm{T}}\boldsymbol{q}_k\boldsymbol{q}_k^{\mathrm{T}}\mathbf{1} \\ &\le (\sqrt{1+\sigma_{\mathrm{u}}} - 1)^2,\end{aligned} \tag{A.82}$$

implying that

$$\|\mathrm{vec}\{\boldsymbol{P}_k - \boldsymbol{p}_k\boldsymbol{p}_k^{\mathrm{T}}\}\|_1 \le \frac{5}{2}\sigma_{\mathrm{u}} + \frac{1}{4}\sigma_{\mathrm{u}}^2, \tag{A.83}$$

which follows from that fact that

$$\sqrt{1+x}-1 \leq \frac{x}{2}$$

holds for all $x \geq 0$. Hence we have

$$\|\Xi_k\|_{\max} \leq \frac{1}{N_{\mathrm{s}}}\left(\frac{5}{2}\sigma_{\mathrm{u}} + \frac{1}{4}\sigma_{\mathrm{u}}^2\right), \tag{A.84}$$

which proves (5.42). Thus the proof is completed. ☐

## A.9 Proof of Proposition 6.1

*Proof.* Consider the transform from $\boldsymbol{\beta}$ to $\boldsymbol{\alpha}$, which helps us to reformulate (6.17) (where the cost function is approximated as in (6.18)) in the form of:

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \xi(\boldsymbol{\alpha}), \\ &\text{s.t. } [\boldsymbol{\alpha}]_1 = 1, \end{aligned} \tag{A.85}$$

where

$$\xi(\boldsymbol{\alpha}) := \tilde{\epsilon}[\boldsymbol{\varphi}(\boldsymbol{\alpha})] = \int_{\lambda_{\mathrm{m}}}^{1} f(\lambda)\sqrt{\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{A}(\lambda)\boldsymbol{\alpha}}\,\mathrm{d}\lambda, \tag{A.86}$$

$\boldsymbol{\varphi}(\cdot)$ is the mapping from $\boldsymbol{\beta}$ to $\boldsymbol{\alpha}$, $\boldsymbol{A}(\lambda)$ is defined by $\boldsymbol{A}(\lambda) := \boldsymbol{a}(\lambda)[\boldsymbol{a}(\lambda)]^{\mathrm{T}}$, and $\boldsymbol{a}(\lambda) := [\lambda^N\ \lambda^{N-1}\ \ldots\ \lambda]^{\mathrm{T}}$. Note that the term $\sqrt{\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{A}(\lambda)\boldsymbol{\alpha}}$ is actually the Mahalanobis norm [215] of $\boldsymbol{\alpha}$ with respect to a positive semi-definite symmetric matrix $\boldsymbol{A}(\lambda)$, hence it is a convex function of $\boldsymbol{\alpha}$. Thus the objective function itself is also convex with respect to $\boldsymbol{\alpha}$, since the integration (weighted by a non-negative function $f(\lambda)$) preserves convexity.

Next, we observe that $\boldsymbol{\varphi}(\cdot)$ can be computed via (6.13), and its inverse may be obtained using the factorization of polynomials [216]. Since $\boldsymbol{\beta}$ satisfies the ordering (6.14), when $\boldsymbol{\alpha}$ is further constrained to be the coefficients of polynomials having only non-negative real-valued roots, it is clear that $\boldsymbol{\varphi}(\cdot)$ is a bijection, and hence the Jacobian $\boldsymbol{J}_{\boldsymbol{\beta}}$ that is given by

$$\boldsymbol{J}_{\boldsymbol{\beta}} = \left[\frac{\partial\boldsymbol{\varphi}(\boldsymbol{\beta})}{\partial\beta_1}\ \frac{\partial\boldsymbol{\varphi}(\boldsymbol{\beta})}{\partial\beta_2}\ \cdots\ \frac{\partial\boldsymbol{\varphi}(\boldsymbol{\beta})}{\partial\beta_{N-1}}\right]^{\mathrm{T}},$$

is invertible for every $\boldsymbol{\beta} \in \mathscr{B}$. This implies that $\boldsymbol{\varphi}(\cdot)$ is a diffeomorphism from $\boldsymbol{\beta}$ to $\boldsymbol{\alpha}$, and hence $\tilde{\epsilon}(\boldsymbol{\beta})$ is an invex function of $\boldsymbol{\beta}$ [192, 217, 218]. To elaborate a little further, we see that

$$\left.\frac{\partial\tilde{\epsilon}(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}}\right|_{\boldsymbol{\beta}_0} = \boldsymbol{J}_{\boldsymbol{\beta}_0}^{-1}\left.\frac{\partial\xi(\boldsymbol{\alpha})}{\partial\boldsymbol{\alpha}}\right|_{\boldsymbol{\varphi}(\boldsymbol{\beta}_0)} = \boldsymbol{0} \Leftrightarrow \left.\frac{\partial\xi(\boldsymbol{\alpha})}{\partial\boldsymbol{\alpha}}\right|_{\boldsymbol{\varphi}(\boldsymbol{\beta}_0)} = \boldsymbol{0}$$

holds for $\boldsymbol{\beta}_0 \in \mathscr{B}$, implying that $\boldsymbol{\beta}_0 \in \mathscr{B}$ is a stationary point of $\tilde{\epsilon}(\boldsymbol{\beta})$ if and only if $\boldsymbol{\varphi}(\boldsymbol{\beta}_0)$ is also a stationary point of $\xi(\boldsymbol{\alpha})$, which in turn is one of the global minima of $\xi(\boldsymbol{\alpha})$.

Our remaining task is to show that $\xi(\boldsymbol{\alpha})$ attains its global minimum when $\boldsymbol{\beta} = \boldsymbol{\varphi}^{-1}(\boldsymbol{\alpha})$ belongs to the feasible region $\mathscr{B}$. This may be proved using the method of contradiction. Assume by contrast that the minimum of $\xi(\boldsymbol{\alpha})$ is attained at $\boldsymbol{\alpha}_0 \notin \mathscr{B}$. Then the polynomial $\boldsymbol{\alpha}_0^{\mathrm{T}}\boldsymbol{a}(\lambda)$ has either real negative roots or complex roots. For the former case, it is plausible that $|\boldsymbol{\alpha}_0^{\mathrm{T}}\boldsymbol{a}(\lambda)| > \lambda^N$ for all $\lambda > 0$, hence $\boldsymbol{\alpha}_0$ is not the optimum. For the latter case, we specifically consider a pair of conjugate complex roots $x \pm iy$. It is clear that

$$\begin{aligned} |(\lambda - x - iy)(\lambda - x + iy)| &= \lambda^2 - 2x\lambda + \sqrt{x^2 + y^2} \\ &\geq \lambda^2 - 2x\lambda + x^2 \\ &= (\lambda - x)^2, \end{aligned}$$

implying that the cost function value can be reduced by replacing the complex roots with real roots. Hence the proof is completed. □

## A.10 Proof of Proposition 6.5

*Proof.* The term $\tilde{\epsilon}(\mathbf{0})$ may be viewed as the $N$-th moment of the Pareto distribution. Upon denoting the shape parameter and the minimum value of the Pareto distribution as $k$ and $\lambda_{\mathrm{m}}$, we have

$$\tilde{\epsilon}(\mathbf{0}) = \frac{k}{k-N} \cdot \lambda_{\mathrm{m}}^N. \tag{A.87}$$

From (6.19) we obtain

$$\begin{aligned} R(\boldsymbol{\beta}) &= \frac{\tilde{\epsilon}(\boldsymbol{\beta})}{\tilde{\epsilon}(\mathbf{0})} \\ &= \frac{k-N}{k\lambda_{\mathrm{m}}^N} \sum_{i=0}^{N-1} (-1)^i \int_{\beta_{N-i-1}}^{\beta_{N-i}} G_{\boldsymbol{\beta}}(\lambda)\mathrm{d}\lambda. \end{aligned} \tag{A.88}$$

Note that for Type-1 permutation filters, we have $\boldsymbol{\beta} = \frac{k\lambda_{\mathrm{m}}}{k-1}\mathbf{1}$. Hence (A.88) can be simplified as

$$\begin{aligned} R(\boldsymbol{\beta}) &= \frac{k-N}{k\lambda_{\mathrm{m}}^N} \left( \int_{\frac{k\lambda_{\mathrm{m}}}{k-1}}^{\infty} |G_{\boldsymbol{\beta}}(\lambda)|\mathrm{d}\lambda + \int_{\lambda_{\mathrm{m}}}^{\frac{k\lambda_{\mathrm{m}}}{k-1}} |G_{\boldsymbol{\beta}}(\lambda)|\mathrm{d}\lambda \right) \\ &= \left| G(\lambda_{\mathrm{m}}) - G\left(\frac{k\lambda_{\mathrm{m}}}{k-1}\right) \right| + \left| G\left(\frac{k\lambda_{\mathrm{m}}}{k-1}\right) \right| \\ &\leq 2\left| G\left(\frac{k\lambda_{\mathrm{m}}}{k-1}\right) \right|, \end{aligned} \tag{A.89}$$

where for simplicity of notations we have defined $G(\lambda) = \frac{k-N}{\lambda_{\mathrm{m}}^{N-k}} \tilde{G}_{\boldsymbol{\alpha}}(\lambda)$. The last line of (A.89) comes from the fact that

$$\int_{\lambda_{\mathrm{m}}}^{\frac{k\lambda_{\mathrm{m}}}{k-1}} |G_{\boldsymbol{\beta}}(\lambda)|\mathrm{d}\lambda \geq 0.$$

Furthermore, from (6.13) we have

$$\alpha_i = \binom{N-1}{i-1}\left(-\frac{k\lambda_\mathrm{m}}{k-1}\right)^{i-1}. \tag{A.90}$$

Thus we obtain

$$\begin{aligned} G(\lambda) &= \sum_{n=1}^{N} \alpha_{N-n+1}\frac{k-N}{n-k}\cdot\frac{\lambda^{n-k}}{\lambda_\mathrm{m}^{N-k}} \\ &= (k-N)\sum_{n=1}^{N}\frac{\binom{N-1}{N-n}}{n-k}\left(\frac{-k}{k-1}\right)^{N-n}\left(\frac{\lambda}{\lambda_\mathrm{m}}\right)^{n-k}. \end{aligned} \tag{A.91}$$

This implies that

$$G\left(\frac{k\lambda_\mathrm{m}}{k-1}\right) = \frac{k-N}{\left(\frac{k}{k-1}\right)^{k-N}}\sum_{n=1}^{N}\frac{\binom{N-1}{n-1}}{n-k}(-1)^{N-n}. \tag{A.92}$$

Next, we denote

$$\sum_{n=1}^{N}(-1)^{N-n}\binom{N-1}{n-1}\eta(n,k) = \boldsymbol{a}_{N-1}^\mathrm{T}\boldsymbol{\eta}, \tag{A.93}$$

where $[\boldsymbol{a}_{N-1}]_i = (-1)^{N-i}\binom{N-1}{i-1}$, $[\boldsymbol{\eta}]_i = \eta(i,k)$, and $\eta(n,k)$ denotes an arbitrary function of $n$ and $k$. Furthermore, we have

$$\boldsymbol{a}_{N-1}^\mathrm{T}\boldsymbol{\eta} = \mathbf{1}^\mathrm{T}\boldsymbol{A}_{N-1}\boldsymbol{\eta}, \tag{A.94}$$

where $\boldsymbol{A}_{N-1}$ is defined recursively by

$$\boldsymbol{A}_n = \begin{bmatrix} \boldsymbol{A}_{n-1} & \mathbf{0}_{2^{n-2}\times 1} \\ \mathbf{0}_{2^{n-2}\times 1} & -\boldsymbol{A}_{n-1} \end{bmatrix}, \tag{A.95}$$

and $\boldsymbol{A}_1 := [1 \ -1]$. Thus we have the following recursion

$$\mathbf{1}^\mathrm{T}\boldsymbol{A}_n\boldsymbol{x} = \mathbf{1}^\mathrm{T}\boldsymbol{A}_{n-1}\left([\boldsymbol{x}]_{1:L-1} - [\boldsymbol{x}]_{2:L}\right)$$

for $\boldsymbol{x} \in \mathbb{R}^L$. From (A.92) we may now write $\boldsymbol{\eta}$ explicitly as

$$\boldsymbol{\eta} = \left[\frac{1}{1-k}\ \frac{1}{2-k}\ \cdots\ \frac{1}{N-k}\right]^\mathrm{T}. \tag{A.96}$$

When $N = 2$, we have

$$\begin{aligned} \mathbf{1}^\mathrm{T}\boldsymbol{A}_1\boldsymbol{\eta} &= \frac{1}{1-k} - \frac{1}{2-k} \\ &= \frac{\Gamma(-k)}{\Gamma(1-k)} - \frac{\Gamma(1-k)}{\Gamma(2-k)}, \end{aligned}$$

where $\Gamma(\cdot)$ denotes the Gamma function [219]. Note that

$$\frac{\Gamma(-k)}{\Gamma(m-k)} - \frac{\Gamma(1-k)}{\Gamma(m-k+1)} = \frac{m\Gamma(-k)}{\Gamma(m+1-k)}. \tag{A.97}$$

Hence for $N = 3$ we obtain

$$\begin{aligned}\mathbf{1}^{\mathrm{T}}\boldsymbol{A}_2\boldsymbol{\eta} &= \frac{\Gamma(-k)}{\Gamma(2-k)} - \frac{\Gamma(1-k)}{\Gamma(2-k+1)} \\ &= \frac{2\Gamma(-k)}{\Gamma(3-k)},\end{aligned}$$

and in general we have

$$\begin{aligned}\mathbf{1}^{\mathrm{T}}\boldsymbol{A}_{N-1}\boldsymbol{\eta} &= \frac{(N-1)!\Gamma(-k)}{\Gamma(N-k)} \\ &= (-1)^N(N-1)! \cdot \frac{\Gamma(k-N-1)}{\Gamma(k)}.\end{aligned}$$

This implies that

$$\begin{aligned}\left|G\left(\frac{k\lambda_{\mathrm{m}}}{k-1}\right)\right| &= \frac{(k-N)(N-1)!}{\left(\frac{k}{k-1}\right)^{k-N}} \cdot \frac{\Gamma(k-N-1)}{\Gamma(k)} \\ &= \frac{\left(1+\frac{1}{k-1}\right)^{N-k}(N-1)!}{\prod_{n=1}^{N-1}(k-n)},\end{aligned} \tag{A.98}$$

as a function of $k$.

Finally, since we have assumed that the spectral density obeys a Pareto distribution, we may compute the relative noise bandwidth explicitly as follows:

$$\begin{aligned}b(\tilde{\boldsymbol{\lambda}}) &= \sqrt{\frac{k}{(k-1)^2(k-2)}} \\ &\geq (k-1)^{-1}.\end{aligned} \tag{A.99}$$

Combining (A.98) and (A.99), we obtain the desired scaling law in (6.42). □

## A.11 Proof of Proposition 6.6

*Proof.* To simplify the discussion, we will use the Pauli basis. Under the Pauli basis, a quantum channel $\mathcal{C}$ may be represented in a matrix form as

$$[\boldsymbol{C}]_{i,j} = \frac{1}{2^{N_{\mathrm{q}}}}\mathrm{Tr}\left\{\mathcal{S}_i\mathcal{C}(\mathcal{S}_j)\right\}, \tag{A.100}$$

where $\mathcal{S}_i$ denotes the $i$-th Pauli string acting upon $N_\mathrm{q}$ qubits. Correspondingly, a quantum state $\rho$ may be represented as a vector:

$$[\boldsymbol{x}_\rho]_i = \frac{1}{\sqrt{2^{N_\mathrm{q}}}} \mathrm{Tr}\left\{\mathcal{S}_i \rho\right\}. \tag{A.101}$$

Since the Pauli operators are unitary and mutually orthogonal, both the transform from the conventional computation basis to the Pauli basis, as well as the inverse transform, are also unitary. This implies that

$$\|\boldsymbol{\lambda}_\rho\|_2 = \|\boldsymbol{x}_\rho\|_\mathrm{F} = \|\boldsymbol{x}_\rho\|_2, \tag{A.102}$$

due to the unitary invariance of the Frobenius norm [214], where $\boldsymbol{\lambda}_\rho$ denotes the vector containing all eigenvalues of $\rho$ sorted in descending order. Without loss of generality, we assume that the first Pauli operator is the identity operator $\mathcal{I}^{\otimes N_\mathrm{q}}$. In light of this, we have

$$\boldsymbol{x}_\rho = [2^{-N_\mathrm{q}/2}\ \tilde{\boldsymbol{x}}_\rho^\mathrm{T}]^\mathrm{T}, \tag{A.103}$$

since all quantum states satisfy $\mathrm{Tr}\left\{\rho\right\} = 1$.

We say that "a layer of gates" is activated if each qubit has been act upon by at least one gate. From our assumption we see that the circuit consists of at least $L$ layers. After the $l$-th layer, the output state $\boldsymbol{x}_{\rho_l}$ may be expressed as

$$\boldsymbol{x}_{\rho_l} = \widetilde{\boldsymbol{G}}_l \boldsymbol{x}_{\rho_{l-1}} = \boldsymbol{C}_l \boldsymbol{G}_l \boldsymbol{x}_{\rho_{l-1}}, \tag{A.104}$$

where $\boldsymbol{G}_l$ denotes the ideal noiseless operation corresponding to the $l$-th layer, and $\boldsymbol{C}_l$ denotes the associated quantum channel characterizing the noise. A perfect layer of gates $\boldsymbol{G}_i$ can be expressed as

$$\boldsymbol{G}_i = \begin{bmatrix} 1 & \boldsymbol{0}^\mathrm{T} \\ \boldsymbol{0} & \boldsymbol{U}_i \end{bmatrix}, \tag{A.105}$$

where $\boldsymbol{U}_i \in \mathbb{R}^{(4^{N_\mathrm{q}}-1)\times(4^{N_\mathrm{q}}-1)}$ is a unitary matrix, while a Pauli channel $\boldsymbol{C}_i$ takes the following form

$$\boldsymbol{C}_i = \begin{bmatrix} 1 & \boldsymbol{0}^\mathrm{T} \\ \boldsymbol{0} & \boldsymbol{D}_i \end{bmatrix}, \tag{A.106}$$

where $\boldsymbol{D}_i \in \mathbb{R}^{(4^{N_\mathrm{q}}-1)\times(4^{N_\mathrm{q}}-1)}$ is a diagonal matrix, whose diagonal entries take values in the interval $[0, 1]$. We now see that the maximum singular value of $\widetilde{\boldsymbol{G}}_l$ is 1, while its second largest singular value $\sigma_2(\widetilde{\boldsymbol{G}}_l)$ is given by

$$\sigma_2(\widetilde{\boldsymbol{G}}_l) = \|\boldsymbol{D}_l\|_2. \tag{A.107}$$

Since the probability of each single-qubit Pauli error is at least $\epsilon_1$, we see that for a single-qubit channel $\mathcal{C}$ characterized by the error probabilities of $p_{\mathrm{X}}$, $p_{\mathrm{Y}}$ and $p_{\mathrm{Z}}$ corresponding to the X, Y and Z errors, respectively, the following holds:

$$\begin{aligned}\boldsymbol{C} &= \mathrm{diag}\left\{\widetilde{\boldsymbol{H}}[1-p_{\mathrm{X}}-p_{\mathrm{Y}}-p_{\mathrm{Z}}\ \ p_{\mathrm{X}}\ \ p_{\mathrm{Y}}\ \ p_{\mathrm{Z}}]^{\mathrm{T}}\right\} \\ &= \boldsymbol{I} - 2\mathrm{diag}\left\{[p_{\mathrm{X}}+p_{\mathrm{Z}}\ \ p_{\mathrm{Y}}+p_{\mathrm{Z}}\ \ p_{\mathrm{X}}+p_{\mathrm{Y}}]\right\} \\ &\preccurlyeq (1-4\epsilon_{\mathrm{l}})\boldsymbol{I},\end{aligned} \tag{A.108}$$

where $\widetilde{\boldsymbol{H}}$ denotes the inverse Hadamard transform over $N_{\mathrm{q}}$ qubits. Therefore, we obtain

$$\begin{aligned}\|\tilde{\boldsymbol{x}}_{\rho_L}\|_2 &\le \sigma_2\left(\prod_{l=1}^{L}\widetilde{\boldsymbol{G}}_{L-l+1}\right) \\ &\le \prod_{l=1}^{L}\|\boldsymbol{D}_l\|_2 \\ &\le (1-4\epsilon_{\mathrm{l}})^L \\ &\le \exp\left(-4\epsilon_{\mathrm{l}}L\right),\end{aligned} \tag{A.109}$$

where the last line follows from the fact that $\ln(1-x) \le -x$ holds for all $x > 0$. This implies that

$$\|\boldsymbol{x}_{\rho_L} - [2^{N_{\mathrm{q}}/2}\ \mathbf{0}^{\mathrm{T}}]^{\mathrm{T}}\|_2 \le \exp\left(-4\epsilon_{\mathrm{l}}L\right). \tag{A.110}$$

Note that $[2^{-N_{\mathrm{q}}/2}\ \mathbf{0}^{\mathrm{T}}]^{\mathrm{T}}$ corresponds to the completely mixed state $2^{-N_{\mathrm{q}}}\boldsymbol{I}$, hence from (A.102) we have

$$\begin{aligned}\|\boldsymbol{\lambda}_{\rho_L} - 2^{-N_{\mathrm{q}}}\mathbf{1}\|_2 &= \|\rho_L - 2^{-N_{\mathrm{q}}}\boldsymbol{I}\|_{\mathrm{F}} \\ &\le \exp\left(-4\epsilon_{\mathrm{l}}L\right).\end{aligned} \tag{A.111}$$

The relative noise bandwidth is given by

$$b(\tilde{\boldsymbol{\lambda}}) = \mu^{-1}(2^{N_{\mathrm{q}}}-1)^{-\frac{1}{2}}\|\tilde{\boldsymbol{\lambda}} - \mu\mathbf{1}\|_2, \tag{A.112}$$

where $\mu = \frac{1-[\boldsymbol{\lambda}_{\rho_L}]_1}{2^{N_{\mathrm{q}}}-1}$ and $\tilde{\boldsymbol{\lambda}} = [\boldsymbol{\lambda}_{\rho_L}]_{2:2^{N_{\mathrm{q}}}}$. The term $\|\tilde{\boldsymbol{\lambda}} - (2^{N_{\mathrm{q}}}-1)^{-1}\mathbf{1}|_2$ can be bounded by

$$\begin{aligned}\|\tilde{\boldsymbol{\lambda}} - \mu\mathbf{1}\|_2 &\le \|\tilde{\boldsymbol{\lambda}} - 2^{-N_{\mathrm{q}}}\mathbf{1}\|_2 + \frac{\left\|[\boldsymbol{\lambda}_{\rho_L}]_1 - 2^{-N_{\mathrm{q}}}\mathbf{1}\right\|_2}{2^{N_{\mathrm{q}}}-1} \\ &\le \left(1 + (2^{N_{\mathrm{q}}}-1)^{-1/2}\right)e^{-4\epsilon_{\mathrm{l}}L}.\end{aligned} \tag{A.113}$$

In addition, we have

$$\begin{aligned}\mu &= \frac{1-2^{-N_{\mathrm{q}}} - |[\boldsymbol{\lambda}_{\rho_L}]_1 - 2^{-N_{\mathrm{q}}}|}{2^{N_{\mathrm{q}}}-1} \\ &\ge \frac{1-2^{-N_{\mathrm{q}}} - e^{-4\epsilon_{\mathrm{l}}L}}{2^{N_{\mathrm{q}}}-1}.\end{aligned} \tag{A.114}$$

Substituting (A.113) and (A.114) into (A.112), we obtain (6.43). Thus the proof is completed. □

## A.12 Proof of Proposition 7.1

*Proof.* When $\omega = |+\rangle\langle+|$, we have

$$\begin{aligned}
&\boldsymbol{C}_{ij}(\rho\otimes\omega)\boldsymbol{C}_{ij}^\dagger \\
&\quad= \frac{1}{2}\left(\boldsymbol{A}_i\boldsymbol{B}_j\otimes|0\rangle\langle0| + \boldsymbol{B}_j\boldsymbol{A}_i\otimes|1\rangle\langle1|\right) \\
&\qquad[\rho\otimes(|0\rangle\langle0| + |0\rangle\langle1| + |1\rangle\langle0| + |1\rangle\langle1|)] \\
&\qquad\left(\boldsymbol{B}_j^\dagger\boldsymbol{A}_i^\dagger\otimes|0\rangle\langle0| + \boldsymbol{A}_i^\dagger\boldsymbol{B}_j^\dagger\otimes|1\rangle\langle1|\right) \\
&\quad= \frac{1}{2}\Big(\boldsymbol{A}_i\boldsymbol{B}_j\rho\boldsymbol{B}_j^\dagger\boldsymbol{A}_i^\dagger\otimes|0\rangle\langle0| + \boldsymbol{A}_i\boldsymbol{B}_j\rho\boldsymbol{A}_i^\dagger\boldsymbol{B}_j^\dagger\otimes \\
&\qquad|0\rangle\langle1| + \boldsymbol{B}_j\boldsymbol{A}_i\rho\boldsymbol{B}_j^\dagger\boldsymbol{A}_i^\dagger\otimes|1\rangle\langle0| \\
&\qquad+ \boldsymbol{B}_j\boldsymbol{A}_i\rho\boldsymbol{A}_i^\dagger\boldsymbol{B}_j^\dagger\otimes|1\rangle\langle1|\Big).
\end{aligned} \tag{A.115}$$

Therefore, if we do not post-select on the control qubit, the partial state on the data register can be obtained by taking the partial trace as

$$\begin{aligned}
&\sum_{i,j}\mathrm{tr}_\omega\{\boldsymbol{C}_{ij}(\rho\otimes\omega)\boldsymbol{C}_{ij}^\dagger\} \\
&\quad= \frac{1}{2}\sum_{i,j}\boldsymbol{A}_i\boldsymbol{B}_j\rho\boldsymbol{B}_j^\dagger\boldsymbol{A}^\dagger + \boldsymbol{B}_j\boldsymbol{A}_i\rho\boldsymbol{A}_i^\dagger\boldsymbol{B}_j^\dagger \\
&\quad= \frac{1}{4}\sum_{i,j}\{\boldsymbol{A}_i,\boldsymbol{B}_j\}\rho\{\boldsymbol{A}_i,\boldsymbol{B}_j\}^\dagger + [\boldsymbol{A}_i,\boldsymbol{B}_j]\rho[\boldsymbol{A}_i,\boldsymbol{B}_j]^\dagger.
\end{aligned} \tag{A.116}$$

But if we discard the computational result once we measure a $|-\rangle$ on the control qubit, note that $\boldsymbol{C}_{ij}(\rho\otimes\omega)\boldsymbol{C}_{ij}^\dagger$ may be further expressed as

$$\begin{aligned}
&\boldsymbol{C}_{ij}(\rho\otimes\omega)\boldsymbol{C}_{ij}^\dagger \\
&= \frac{1}{4}\Big[\boldsymbol{A}_i\boldsymbol{B}_j\rho\boldsymbol{B}_j^\dagger\boldsymbol{A}_i^\dagger\otimes(|+\rangle\langle+| + |+\rangle\langle-| + |-\rangle\langle+| + |-\rangle\langle-|) \\
&\quad+\boldsymbol{A}_i\boldsymbol{B}_j\rho\boldsymbol{A}_i^\dagger\boldsymbol{B}_j^\dagger\otimes(|+\rangle\langle+| - |+\rangle\langle-| + |-\rangle\langle+| - |-\rangle\langle-|) \\
&\quad+\boldsymbol{B}_j\boldsymbol{A}_i\rho\boldsymbol{B}_j^\dagger\boldsymbol{A}_i^\dagger\otimes(|+\rangle\langle+| + |+\rangle\langle-| - |-\rangle\langle+| - |-\rangle\langle-|) \\
&\quad+\boldsymbol{B}_j\boldsymbol{A}_i\rho\boldsymbol{A}_i^\dagger\boldsymbol{B}_j^\dagger\otimes(|+\rangle\langle+| - |+\rangle\langle-| - |-\rangle\langle+| + |-\rangle\langle-|)\Big] \\
&= \frac{1}{4}\Big[\{\boldsymbol{A}_i,\boldsymbol{B}_j\}\rho\{\boldsymbol{A}_i,\boldsymbol{B}_j\}^\dagger\otimes|+\rangle\langle+| \\
&\quad+\{\boldsymbol{A}_i,\boldsymbol{B}_j\}\rho[\boldsymbol{A}_i,\boldsymbol{B}_j]^\dagger\otimes|+\rangle\langle-| \\
&\quad+[\boldsymbol{A}_i,\boldsymbol{B}_j]\rho\{\boldsymbol{A}_i,\boldsymbol{B}_j\}^\dagger\otimes|-\rangle\langle+| \\
&\quad+[\boldsymbol{A}_i,\boldsymbol{B}_j]\rho[\boldsymbol{A}_i,\boldsymbol{B}_j]^\dagger\otimes|-\rangle\langle-|\Big].
\end{aligned} \tag{A.117}$$

Hence the data register will be in the following state

$$\frac{1}{Z}\sum_{i,j}\frac{\{\boldsymbol{A}_i,\boldsymbol{B}_j\}}{2}\rho\frac{\{\boldsymbol{A}_i,\boldsymbol{B}_j\}^\dagger}{2}, \tag{A.118}$$

where $Z$ is a normalization factor given by

$$Z=\frac{\mathrm{Tr}\{\sum_{i,j}\{\boldsymbol{A}_i,\boldsymbol{B}_j\}\rho\{\boldsymbol{A}_i,\boldsymbol{B}_j\}^\dagger+[\boldsymbol{A}_i,\boldsymbol{B}_j]\rho[\boldsymbol{A}_i,\boldsymbol{B}_j]^\dagger\}}{\mathrm{Tr}\{\sum_{i,j}\{\boldsymbol{A}_i,\boldsymbol{B}_j\}\rho\{\boldsymbol{A}_i,\boldsymbol{B}_j\}^\dagger\}}.$$

This completes the proof. □

# Appendix B

# Supplementary Illustrations

## B.1 The Values of the Quasi-probability Representation Vector For Basic Pauli Channels

To elaborate further on the intuition about $\eta_C$, the values of $\eta_C$ corresponding to some single-qubit Pauli channels are listed in Table B.1.

TABLE B.1
$\eta_C$ values of single-qubit Pauli channels.

| Channel | Parameters | $\eta_C$ |
|---|---|---|
| Depolarizing | Depolarizing prob. $p$ | $[1-p,\ \frac{p}{3},\ \frac{p}{3},\ \frac{p}{3}]^{\mathrm{T}}$ |
| Bit-flip | Bit-flip prob. $p_{\mathrm{x}}$ | $[1-p_{\mathrm{x}},\ p_{\mathrm{x}},\ 0,\ 0]^{\mathrm{T}}$ |
| Phase-flip | Phase-flip prob. $p_{\mathrm{z}}$ | $[1-p_{\mathrm{z}},\ 0,\ 0,\ p_{\mathrm{z}}]^{\mathrm{T}}$ |

## B.2 Notes on the Spectral Response of Permutation Filters

Let us consider a third-order permutation filter as an example, which has the following spectral response:

$$h_{\beta}(\lambda) = \lambda(\lambda - \beta_1)(\lambda - \beta_2), \tag{B.1}$$

where $\beta_1 \leq \beta_2$. By taking the limit $\lambda \to \infty$, we see that $h_{\beta}(\lambda) \sim \lambda^3$, implying that the spectral response can be well approximated by $\lambda^3$ when $\lambda \gg \beta_2$. Since the cubic function $\lambda^3$ satisfies $\lambda_1^3 = 10^3 \cdot \lambda_2^3$ when $\lambda_1 = 10\lambda_2$, we say that it "has a slope of 30dB per decade" (note that 10dB corresponds to $10\log_{10}(10) = 10$ times). Here, the "slope" refers to that of the spectral response curve on a log-log scale, which appears to be linear for power functions. Furthermore, if $\beta_1$ and $\beta_2$ is well separated, we see that $h_{\beta}(\lambda) \sim \lambda^2$ when $\beta_1 \ll \lambda \ll \beta_2$, and hence "has a slope of 20dB per decade". In general, when the eigenvalue $\lambda$ is in the region $\beta_n \ll \lambda \ll \beta_{n+1}$, we see that the slope is (approximately) $10(n+1)$ dB per decade. Since the first zero is $\beta_0 = 0$, we may conclude that each zero $\beta_i \ll \lambda_0$ contributes 10dB/decade to the slope at the point $\lambda = \lambda_0$.

# References

[1] Y. Xiong, D. Chandra, S. X. Ng, and L. Hanzo, "Sampling overhead analysis of quantum error mitigation: Uncoded vs. coded systems," *IEEE Access*, vol. 8, pp. 228967–228991, 2020.

[2] Y. Xiong, S. X. Ng, and L. Hanzo, "Quantum error mitigation relying on permutation filtering," *IEEE Transactions on Communications*, 2021, *early access*.

[3] Y. Xiong, S. X. Ng, and L. Hanzo, "The accuracy vs. sampling overhead trade-off in quantum error mitigation using monte carlo-based channel inversion," *IEEE Transactions on Communications*, 2022, *early access*.

[4] Y. Xiong, S. X. Ng, G.-L. Long, and L. Hanzo, "Dual-frequency quantum phase estimation mitigates the spectral leakage of quantum algorithms," *IEEE Signal Process. Lett.*, vol. 29, pp. 1222–1226, Apr. 2022.

[5] Y. Xiong, D. Chandra, S. X. Ng, and L. Hanzo, "Circuit-oriented symmetry verification: From quantum switches to spatio-temporal stabilizers," *arXiv preprint*, 2021.

[6] C. E. Leiserson, R. L. Rivest, T. H. Cormen, and C. Stein, *Introduction to algorithms*, vol. 6. Cambridge, MA, USA: MIT Press, 2001.

[7] F. Arute, K. Arya, R. Babbush, *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, pp. 505–510, Oct. 2019.

[8] S. Aaronson and A. Arkhipov, "The computational complexity of linear optics," in *Proc. 43rd annual ACM symposium on Theory of computing*, (San Jose, CA, USA), pp. 333–342, June 2011.

[9] P. Zialcita, "Google claims to achieve quantum supremacy—IBM pushes back," Oct. 2019.

[10] R. P. Feynman, "Simulating physics with computers," *International Journal of Theoretical Physics*, vol. 21, pp. 467–488, June 1982.

[11] G. H. Low and I. L. Chuang, "Optimal Hamiltonian simulation by quantum signal processing," *Physical Review Letters*, vol. 118, p. 010501, Jan. 2017.

[12] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annual Symp. Foundations of Computer Science*, (Santa Fe, New Mexico, USA), pp. 124–134, Nov. 1994.

[13] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Annual ACM Symp. Theory of Computing*, (Philadelphia, Pennsylvania, USA), pp. 212–219, May 1996.

[14] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, "Tight bounds on quantum searching," *Fortschritte der Physik: Progress of Physics*, vol. 46, pp. 493–505, June 1998.

[15] C. Durr and P. Høyer, "A quantum algorithm for finding the minimum," *arXiv preprint*, 1996.

[16] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Physical Review Letters*, vol. 103, Oct. 2009.

[17] P. Botsinis, S. X. Ng, and L. Hanzo, "Quantum search algorithms, quantum wireless, and a low-complexity maximum likelihood iterative quantum multi-user detector design," *IEEE Access*, vol. 1, pp. 94–122, 2013.

[18] D. Alanis, P. Botsinis, S. X. Ng, and L. Hanzo, "Quantum-assisted routing optimization for self-organizing networks," *IEEE Access*, vol. 2, pp. 614–632, 2014.

[19] D. Alanis, P. Botsinis, Z. Babar, S. X. Ng, and L. Hanzo, "Non-dominated quantum iterative routing optimization for wireless multihop networks," *IEEE Access*, vol. 3, pp. 1704–1728, 2015.

[20] P. Botsinis, D. Alanis, Z. Babar, H. V. Nguyen, D. Chandra, S. X. Ng, and L. Hanzo, "Quantum search algorithms for wireless communications," *IEEE Commun. Surv. Tuts.*, vol. 21, pp. 1209–1242, 2nd Quart. 2019.

[21] E. Schrödinger, "Die gegenwärtige situation in der quantenmechanik," *Naturwissenschaften*, vol. 23, no. 49, pp. 807–812, 1935.

[22] C. Rigetti, J. M. Gambetta, S. Poletto, B. L. T. Plourde, J. M. Chow, A. D. Córcoles, J. A. Smolin, S. T. Merkel, J. R. Rozen, G. A. Keefe, *et al.*, "Superconducting qubit in a waveguide cavity with a coherence time approaching 0.1 ms," *Physical Review B*, vol. 86, p. 100506, Sept. 2012.

[23] D. Aharonov and M. Ben-Or, "Fault-tolerant quantum computation with constant error rate," *SIAM Journal on Computing*, vol. 38, p. 1207–1282, July 2008.

[24] R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, "Quantum error correction via codes over GF(4)," *IEEE Transactions on Information Theory*, vol. 44, pp. 1369–1387, July 1998.

[25] C. H. Bennett and P. W. Shor, "Quantum information theory," *IEEE Transactions on Information Theory*, vol. 44, pp. 2724–2742, Oct. 1998.

[26] Z. Babar, D. Chandra, H. V. Nguyen, P. Botsinis, D. Alanis, S. X. Ng, and L. Hanzo, "Duality of quantum and classical error correction codes: Design principles and examples," *IEEE Commun. Surv. Tuts.*, vol. 21, pp. 970–1010, 1st quart. 2019.

[27] D. Chandra, Z. Babar, H. V. Nguyen, D. Alanis, P. Botsinis, S. X. Ng, and L. Hanzo, "Quantum topological error correction codes: The classical-to-quantum isomorphism perspective," *IEEE Access*, vol. 6, pp. 13729–13757, 2018.

[28] E. Knill and R. Laflamme, "Concatenated quantum codes," *arXiv preprint*, 1996.

[29] D. Poulin, J. Tillich, and H. Ollivier, "Quantum serial turbo codes," *IEEE Transactions on Information Theory*, vol. 55, pp. 2776–2798, June 2009.

[30] Z. Babar, S. X. Ng, and L. Hanzo, "EXIT-chart-aided near-capacity quantum turbo code design," *IEEE Transactions on Vehicular Technology*, vol. 64, pp. 866–875, Mar. 2015.

[31] J. Preskill, "Quantum Computing in the NISQ era and beyond," *Quantum*, vol. 2, pp. 1–21, Aug. 2018.

[32] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint*, 2014.

[33] S. Hadfield, Z. Wang, B. O'Gorman, E. Rieffel, D. Venturelli, and R. Biswas, "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz," *Algorithms*, vol. 12, pp. 1–45, Feb. 2019.

[34] P. J. Love, J. L. O'Brien, A. Aspuru-Guzik, A. Peruzzo, M.-H. Yung, X.-Q. Zhou, P. Shadbolt, and J. McClean, "A variational eigenvalue solver on a photonic quantum processor," *Nature Commun.*, vol. 5, pp. 1–7, July 2014.

[35] J. I. Colless, V. V. Ramasesh, D. Dahlen, M. S. Blok, M. E. Kimchi-Schwartz, J. McClean, J. Carter, W. A. de Jong, and I. Siddiqi, "Computation of molecular spectra on a quantum processor with an error-resilient algorithm," *Phys.l Rev. X*, vol. 8, Feb. 2018.

[36] X. Bonet-Monroig, R. Sagastizabal, M. Singh, and T. E. O'Brien, "Low-cost error mitigation by symmetry verification," *Physical Review A*, vol. 98, pp. 1–10, Dec. 2018.

[37] K. Temme, S. Bravyi, and J. M. Gambetta, "Error mitigation for short-depth quantum circuits," *Physical Review Letters*, vol. 119, pp. 1–5, Nov. 2017.

[38] S. Endo, S. Benjamin, and Y. Li, "Practical quantum error mitigation for near-future applications," *Physical Review X*, vol. 8, pp. 1–21, July 2018.

[39] C. Song, J. Cui, H. Wang, J. Hao, H. Feng, and Y. Li, "Quantum computation with universal error mitigation on a superconducting quantum processor," *Science Advances*, vol. 5, no. 9, 2019.

[40] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Physical Review A*, vol. 52, p. R2493, Oct. 1995.

[41] A. Steane, "Error correcting codes in quantum theory," *Physical Review Letters*, vol. 77, pp. 793–797, July 1996.

[42] R. Laflamme, C. Miquel, J. P. Paz, and W. H. Zurek, "Perfect quantum error correcting code," *Physical Review Letters*, vol. 77, p. 198, July 1996.

[43] D. Gottesman, "Class of quantum error-correcting codes saturating the quantum hamming bound," *Physical Review A*, vol. 54, p. 1862, Sept. 1996.

[44] E. Knill, R. Laflamme, and W. H. Zurek, "Resilient quantum computation: Error models and thresholds," *Proc. Roy.l Soc. London A, Math. Phys. Eng. Sci.*, vol. 454, p. 365–384, Jan. 1998.

[45] J. Preskill, "Reliable quantum computers," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, pp. 385–410, Jan. 1998.

[46] B. Eastin and E. Knill, "Restrictions on transversal encoded quantum gate sets," *Physical Review Letters*, vol. 102, p. 110502, Mar. 2009.

[47] S. Bravyi and A. Vargo, "Simulation of rare events in quantum error correction," *Physical Review A*, vol. 88, p. 062308, Dec. 2013.

[48] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. New York, NY, USA: Cambridge University Press, 2nd ed., 2011.

[49] A. Kitaev, "Quantum computations: Algorithms and error correction," *Uspekhi Matematicheskikh Nauk*, vol. 52, no. 6, pp. 53–112, 1997.

[50] A. Kitaev, "Fault-tolerant quantum computation by anyons," *Annals of Physics*, vol. 303, pp. 2–30, Jan. 2003.

[51] D. Chandra, Z. Babar, H. V. Nguyen, D. Alanis, P. Botsinis, S. X. Ng, and L. Hanzo, "Quantum topological error correction codes are capable of improving the performance of clifford gates," *IEEE Access*, vol. 7, pp. 121501–121529, 2019.

[52] D. Gottesman, *Stabilizer codes and quantum error correction*. PhD dissertation, California Institute of Technology, 1997.

[53] S. Bravyi, G. Duclos-Cianci, D. Poulin, and M. Suchara, "Subsystem surface codes with three-qubit check operators," *Quantum Information & Computation*, vol. 13, pp. 963–985, Nov. 2013.

[54] H. Bombín, "Topological subsystem codes," *Physical Review A*, vol. 81, p. 032301, Mar. 2010.

[55] M. B. Hastings and J. Haah, "Dynamically generated logical qubits," *Quantum*, vol. 5, p. 564, 2021.

[56] C. Vuillot, "Planar Floquet codes," *arXiv preprint*, 2021.

[57] W. Cook and A. Rohe, "Computing minimum-weight perfect matchings," *INFORMS journal on computing*, vol. 11, pp. 138–148, May 1999.

[58] D. Chandra, Z. Babar, H. V. Nguyen, D. Alanis, P. Botsinis, S. X. Ng, and L. Hanzo, "Quantum topological error correction codes are capable of improving the performance of Clifford gates," *IEEE Access*, vol. 7, pp. 121501–121529, Aug. 2019.

[59] H. Bombin and M. A. Martin-Delgado, "Topological quantum distillation," *Physical Review Letters*, vol. 97, p. 180501, Oct. 2006.

[60] "IBM quantum experience."

[61] V. V. Shende, S. S. Bullock, and I. L. Markov, "Synthesis of quantum-logic circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, pp. 1000–1010, June 2006.

[62] J. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," *New Journal of Physics*, vol. 18, pp. 1–22, Feb. 2016.

[63] S. Endo, Q. Zhao, Y. Li, S. Benjamin, and X. Yuan, "Mitigating algorithmic errors in a Hamiltonian simulation," *Physical Review A*, vol. 99, p. 012334, Jan. 2019.

[64] P. Czarnik, A. Arrasmith, P. J. Coles, and L. Cincio, "Error mitigation with Clifford quantum-circuit data," *Quantum*, vol. 5, p. 592, 2021.

[65] B. Koczor, "Exponential error suppression for near-term quantum devices," *Physical Review X*, vol. 11, p. 031057, Sept. 2021.

[66] W. J. Huggins, S. McArdle, T. E. O'Brien, J. Lee, N. C. Rubin, S. Boixo, K. B. Whaley, R. Babbush, and J. McClean, "Virtual distillation for quantum error mitigation," *Physical Review X*, vol. 11, p. 041036, Nov. 2021.

[67] S. McArdle, S. Endo, A. Aspuru-Guzik, S. Benjamin, and X. Yuan, "Quantum computational chemistry," *Reviews of Modern Physics*, vol. 92, p. 015003, Mar. 2020.

[68] R. Sweke, F. Wilde, J. J. Meyer, M. Schuld, P. K. Fährmann, B. Meynard-Piganeau, and J. Eisert, "Stochastic gradient descent for hybrid quantum-classical optimization," *Quantum*, vol. 4, p. 314, 2020.

[69] D. Zhu, N. M. Linke, M. Benedetti, K. A. Landsman, N. H. Nguyen, C. H. Alderete, A. Perdomo-Ortiz, N. Korda, A. Garfoot, C. Brecque, *et al.*, "Training of quantum circuits on a hybrid quantum computer," *Science advances*, vol. 5, p. eaaw9918, Oct. 2019.

[70] Z.-C. Yang, A. Rahmani, A. Shabani, H. Neven, and C. Chamon, "Optimizing variational quantum algorithms using pontryagin's minimum principle," *Physical Review X*, vol. 7, p. 021027, June 2017.

[71] T. Matsumine, T. Koike-Akino, and Y. Wang, "Channel decoding with quantum approximate optimization algorithm," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, (Paris, France), pp. 2574–2578, July 2019.

[72] E. Anschuetz, J. Olson, A. Aspuru-Guzik, and Y. Cao, "Variational quantum factoring," in *Proc. Int. Workshop on Quantum Technology and Optimization Problems*, (Garching b. München, German), pp. 74–85, Springer, Mar. 2019.

[73] G. Verdon, J. Pye, and M. Broughton, "A universal training algorithm for quantum deep learning," *arXiv preprint*, 2018.

[74] S. Endo, Z. Cai, S. Benjamin, and X. Yuan, "Hybrid quantum-classical algorithms and quantum error mitigation," *Journal of the Physical Society of Japan*, vol. 90, p. 032001, Feb. 2021.

[75] H. F. Trotter, "On the product of semi-groups of operators," *Proc. American Math. Soc.*, vol. 10, pp. 545–551, Aug. 1959.

[76] Y. Aharonov, J. Anandan, S. Popescu, and L. Vaidman, "Superpositions of time evolutions of a quantum system and a quantum time-translation machine," *Physical Review Letters*, vol. 64, pp. 2965–2968, June 1990.

[77] G. Chiribella, G. M. D'Ariano, P. Perinotti, and B. Valiron, "Quantum computations without definite causal structure," *Physical Review A*, vol. 88, p. 022318, Aug. 2013.

[78] K. Goswami, C. Giarmatzi, M. Kewming, F. Costa, C. Branciard, J. Romero, and A. G. White, "Indefinite causal order in a quantum switch," *Physical Rev. Lett.*, vol. 121, p. 090503, Aug. 2018.

[79] M. Caleffi and A. S. Cacciapuoti, "Quantum switch for the quantum internet: Noiseless communications through noisy channels," *IEEE Journal on Selected Areas in Communications*, vol. 38, pp. 575–588, 3rd Quart. 2020.

[80] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

[81] H. Zinkernagel, "Niels Bohr on the wave function and the classical/quantum divide," *Studies in History and Philosophy of Science Part B: Studies in History and Philosophy of Modern Physics*, vol. 53, pp. 9–19, Feb. 2016.

[82] N. Harrigan and R. W. Spekkens, "Einstein, incompleteness, and the epistemic view of quantum states," *Foundations of Physics*, vol. 40, pp. 125–157, Jan. 2010.

[83] A. Einstein, B. Podolsky, and N. Rosen, "Can quantum-mechanical description of physical reality be considered complete?," *Physical Review*, vol. 47, pp. 777–780, May 1935.

[84] M. F. Pusey, J. Barrett, and T. Rudolph, "On the reality of the quantum state," *Nature Physics*, vol. 8, pp. 475–478, May 2012.

[85] J. S. Bell, "On the einstein podolsky rosen paradox," *Physics Physique Fizika*, vol. 1, pp. 195–200, Nov. 1964.

[86] D. J. Griffiths and D. F. Schroeter, *Introduction to quantum mechanics*. Cambridge University Press, 2018.

[87] D. Bump, *Lie groups*. Springer, 2004.

[88] J. Daintith and E. Wright, *A Dictionary of Computing*. USA: Oxford University Press, Inc., 6th ed., 2010.

[89] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, "Exponential algorithmic speedup by a quantum walk," in *Proc. 35th annual ACM symposium on Theory of computing*, (San Diego, CA, USA), pp. 59–68, June 2003.

[90] A. M. Childs, L. J. Schulman, and U. Vazirani, "Quantum algorithms for hidden nonlinear structures," in *Proc. 2017 48th Annual IEEE Symposium on Foundations of Computer Science*, (Providence, RI, USA), pp. 395–404, IEEE, Oct. 2007.

[91] A. Kitaev, "Quantum measurements and the abelian stabilizer problem," *arXiv preprint*, 1995.

[92] S. M. Kay, *Fundamentals of statistical signal processing: Estimation theory*. Prentice-Hall, Inc., 1st ed., 1993.

[93] W. Górecki, R. Demkowicz-Dobrzański, H. M. Wiseman, and D. W. Berry, "$\pi$-corrected Heisenberg limit," *Physical Review Letters*, vol. 124, p. 030501, Jan. 2020.

[94] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, "Quantum amplitude amplification and estimation," *Contemporary Mathematics*, vol. 305, pp. 53–74, 2002.

[95] D. W. Berry and A. M. Childs, "Black-box Hamiltonian simulation and unitary implementation," *Quantum Info. Comput.*, vol. 12, p. 29–62, jan 2012.

[96] A. M. Childs, "On the relationship between continuous-and discrete-time quantum walk," *Communications in Mathematical Physics*, vol. 294, pp. 581–603, Mar. 2010.

[97] D. W. Berry, A. M. Childs, and R. Kothari, "Hamiltonian simulation with nearly optimal dependence on all parameters," in *Proc. 2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, (Berkely, CA, USA), pp. 792–809, IEEE, Oct. 2015.

[98] P. Wocjan and S. Zhang, "Several natural BQP-complete problems," *arXiv preprint*, 2006.

[99] S. Aaronson, "BQP and the polynomial hierarchy," in *Proc. 42nd ACM symposium on Theory of computing*, (Cambridge, MA, USA), pp. 141–150, June 2010.

[100] E. Bernstein and U. Vazirani, "Quantum complexity theory," *SIAM Journal on Computing*, vol. 26, pp. 1411–1473, Oct. 1997.

[101] D.-Z. Du and K.-I. Ko, *Theory of computational complexity*, vol. 58. John Wiley & Sons, 2011.

[102] O. Higgott, D. Wang, and S. Brierley, "Variational quantum computation of excited states," *Quantum*, vol. 3, July 2019.

[103] T. Jones, S. Endo, S. McArdle, X. Yuan, and S. Benjamin, "Variational quantum algorithms for discovering Hamiltonian spectra," *Physical Review A*, vol. 99, June 2019.

[104] N. Moll, P. Barkoutsos, L. S. Bishop, *et al.*, "Quantum optimization using variational algorithms on near-term quantum devices," *Quantum Science and Technology*, vol. 3, pp. 1–17, June 2018.

[105] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, pp. 308–313, Jan. 1965.

[106] P. Botsinis, D. Alanis, Z. Babar, H. V. Nguyen, D. Chandra, S. X. Ng, and L. Hanzo, "Quantum search algorithms for wireless communications," *IEEE Commun. Surv. Tuts.*, vol. 21, pp. 1209–1242, 2nd quart. 2019.

[107] D. Greenbaum, “Introduction to quantum gate set tomography,” *arXiv preprint*, 2015.

[108] E. Huang, A. C. Doherty, and S. Flammia, “Performance of quantum error correction with coherent errors,” *Physical Review A*, vol. 99, p. 022313, Feb. 2019.

[109] D. Greenbaum and Z. Dutton, “Modeling coherent errors in quantum error correction,” *Quantum Science and Technology*, vol. 3, p. 015007, Dec. 2017.

[110] A. Katabarwa, “A dynamical interpretation of the Pauli twirling approximation and quantum error correction,” *arXiv preprint*, 2017.

[111] Z. Babar, P. Botsinis, D. Alanis, S. X. Ng, and L. Hanzo, “The road from classical to quantum codes: A hashing bound approaching design procedure,” *IEEE Access*, vol. 3, pp. 146–176, Mar. 2015.

[112] R. Calderbank and P. W. Shor, “Good quantum error-correcting codes exist,” *Physical Review A*, vol. 54, pp. 1098–1105, Aug. 1996.

[113] A. Steane, “Multiple-particle interference and quantum error correction,” *Proc. Royal Soc. London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 452, pp. 2551–2577, Nov. 1996.

[114] M. Freedman, A. Kitaev, M. Larsen, and Z. Wang, “Topological quantum computation,” *Bulletin of the American Mathematical Society*, vol. 40, no. 1, pp. 31–38, 2003.

[115] S. Bravyi and A. Kitaev, “Quantum codes on a lattice with boundary,” *arXiv preprint*, 1998.

[116] G. Smith and J. A. Smolin, “Degenerate quantum codes for pauli channels,” *Physical Review Letters*, vol. 98, p. 030501, Jan. 2007.

[117] D. Cuomo, M. Caleffi, K. Krsulich, F. Tramonto, G. Agliardi, E. Prati, and A. S. Cacciapuoti, “Optimized compiler for distributed quantum computing,” *arXiv preprint*, 2021.

[118] Z. Cai, “Multi-exponential error extrapolation and combining error mitigation techniques for NISQ applications,” *npj Quantum Information*, vol. 7, pp. 1–12, May 2021.

[119] Y. Suzuki, S. Endo, K. Fujii, and Y. Tokunaga, “Quantum error mitigation for fault-tolerant quantum computing,” *arXiv preprint*, 2020.

[120] A. Strikis, D. Qin, Y. Chen, S. Benjamin, and Y. Li, “Learning-based quantum error mitigation,” *PRX Quantum*, vol. 2, p. 040330, Nov. 2021.

[121] A. Lowe, M. H. Gordon, P. Czarnik, A. Arrasmith, P. J. Coles, and L. Cincio, “Unified approach to data-driven quantum error mitigation,” *Physical Review Research*, vol. 3, p. 033098, July 2021.

[122] R. Sagastizabal, X. Bonet-Monroig, M. Singh, M. A. Rol, C. C. Bultink, X. Fu, C. H. Price, V. P. Ostroukh, N. Muthusubramanian, A. Bruno, *et al.*, “Experimental error mitigation via symmetry verification in a variational quantum eigensolver,” *Physical Review A*, vol. 100, p. 010302, Jan. 2019.

[123] B. Koczor, "The dominant eigenvector of a noisy quantum state," *New Journal of Physics*, Nov. 2021.

[124] M. Mohseni, A. T. Rezakhani, and D. A. Lidar, "Quantum-process tomography: Resource analysis of different strategies," *Physical Review A*, vol. 77, pp. 1–15, Mar. 2008.

[125] Z. Cai and S. Benjamin, "Constructing smaller pauli twirling sets for arbitrary error channels," *Scientific reports*, vol. 9, pp. 1–11, Aug. 2019.

[126] C. Dankert, R. Cleve, J. Emerson, and E. Livine, "Exact and approximate unitary 2-designs and their application to fidelity estimation," *Physical Review A*, vol. 80, July 2009.

[127] J. McClean, Z. Jiang, N. C. Rubin, R. Babbush, and H. Neven, "Decoding quantum errors with subspace expansions," *Nat. Commun.*, vol. 11, no. 1, pp. 1–9, 2020.

[128] S. McArdle, X. Yuan, and S. Benjamin, "Error-mitigated digital quantum simulation," *Physical Review Letters*, vol. 122, p. 180501, May 2019.

[129] Z. Cai, "Quantum error mitigation using symmetry expansion," *Quantum*, vol. 5, Sept. 2021.

[130] P. Czarnik, A. Arrasmith, L. Cincio, and P. J. Coles, "Qubit-efficient exponential suppression of errors," *arXiv preprint*, 2021.

[131] Z. Cai, "Resource-efficient purification-based quantum error mitigation," *arXiv preprint*, 2021.

[132] R. B. Patel, J. Ho, F. Ferreyrol, T. C. Ralph, and G. J. Pryde, "A quantum Fredkin gate," *Science Advances*, vol. 2, p. e1501531, Mar. 2016.

[133] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD, USA: Johns Hopkins University Press, 3rd ed., 1996.

[134] A. Kandala, K. Temme, A. D. Córcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, "Error mitigation extends the computational reach of a noisy quantum processor," *Nature*, vol. 567, pp. 491–495, Mar. 2019.

[135] L. F. Richardson and J. A. Gaunt, "VIII. The deferred approach to the limit," *Phil. Trans. R. Soc. A*, vol. 226, pp. 299–361, Jan. 1927.

[136] T. Giurgica-Tiron, Y. Hindy, R. LaRose, A. Mari, and W. J. Zeng, "Digital zero noise extrapolation for quantum error mitigation," in *Proc. 2020 IEEE Int. Conf. Quantum Comput. Engineering (QCE)*, (Virtual Conference), pp. 306–316, Oct. 2020.

[137] D. Manzano, "A short introduction to the Lindblad master equation," *AIP Advances*, vol. 10, p. 025106, Feb. 2020.

[138] J. W. O. Garmon, R. C. Pooser, and E. F. Dumitrescu, "Benchmarking noise extrapolation with the openpulse control framework," *Physical Review A*, vol. 101, p. 042308, Apr. 2020.

[139] E. F. Dumitrescu, A. J. McCaskey, G. Hagen, G. R. Jansen, T. D. Morris, T. Papenbrock, R. C. Pooser, D. J. Dean, and P. Lougovski, "Cloud quantum computing of an atomic nucleus," *Physical Review Letters*, vol. 120, p. 210501, May 2018.

[140] A. He, B. Nachman, W. A. de Jong, and C. Bauer, "Zero-noise extrapolation for quantum-gate error mitigation with identity insertions," *Physical Review A*, vol. 102, p. 012426, July 2020.

[141] C. Bauer, A. He, W. A. de Jong, B. Nachman, and V. R. Pascuzzi, "Computationally efficient zero noise extrapolation for quantum gate error mitigation," *arXiv preprint*, 2021.

[142] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Learning*, vol. 35, pp. 1798–1828, Aug. 2013.

[143] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Learning*, vol. 38, pp. 295–307, Feb. 2016.

[144] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient $k$-means clustering algorithm: Analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Learning*, vol. 24, pp. 881–892, July 2002.

[145] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Learning*, vol. 39, pp. 640–651, Apr. 2017.

[146] D. Gottesman, "Theory of fault-tolerant quantum computation," *Physical Review A*, vol. 57, no. 1, p. 127, 1998.

[147] S. Aaronson and D. Gottesman, "Improved simulation of stabilizer circuits," *Physical Review A*, vol. 70, p. 052328, Nov. 2004.

[148] Y. Yang, B.-N. Lu, and Y. Li, "Accelerated quantum Monte Carlo with mitigated error on noisy quantum computer," *PRX Quantum*, vol. 2, p. 040361, Dec. 2021.

[149] J. A. Barker, "A quantum-statistical monte carlo method; path integrals with boundary conditions," *The Journal of Chemical Physics*, vol. 70, no. 6, pp. 2914–2918, 1979.

[150] D. S. Abrams and S. Lloyd, "Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors," *Physical Review Letters*, vol. 83, p. 5162, Dec. 1999.

[151] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, "Quantum algorithms revisited," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, pp. 339–354, Jan. 1998.

[152] M. A. Nielsen, "A simple formula for the average gate fidelity of a quantum dynamical operation," *Phys. Lett. A*, vol. 303, pp. 249 – 252, Oct. 2002.

[153] A. S. Fletcher, P. W. Shor, and M. Z. Win, "Channel-adapted quantum error correction for the amplitude damping channel," *IEEE Transactions on Information Theory*, vol. 54, pp. 5705–5718, Dec. 2008.

[154] A. Carignan-Dugas, M. Alexander, and J. Emerson, "A polar decomposition for quantum channels (with applications to bounding error propagation in quantum circuits)," *Quantum*, vol. 3, p. 173, Aug. 2019.

[155] F. R. K. Chung and F. C. Graham, *Spectral graph theory*. No. 92, American Mathematical Soc., 1997.

[156] D. J. C. MacKay, G. Mitchison, and P. L. McFadden, "Sparse-graph codes for quantum error correction," *IEEE Transactions on Information Theory*, vol. 50, pp. 2315–2330, Oct. 2004.

[157] M. M. Wilde, M. Hsieh, and Z. Babar, "Entanglement-assisted quantum turbo codes," *IEEE Transactions on Information Theory*, vol. 60, pp. 1203–1222, Feb. 2014.

[158] D. Chandra, Z. Babar, S. X. Ng, and L. Hanzo, "Near-hashing-bound multiple-rate quantum turbo short-block codes," *IEEE Access*, vol. 7, pp. 52712–52730, 2019.

[159] D. P. Palomar and Y. Jiang, *MIMO transceiver design via majorization theory*. Now Publishers Inc., 2007.

[160] T. Ando, "Majorization, doubly stochastic matrices, and comparison of eigenvalues," *Linear Algebra and its Applications*, vol. 118, pp. 163–248, June 1989.

[161] D. P. DiVincenzo, D. W. Leung, and B. M. Terhal, "Quantum data hiding," *IEEE Transactions on Information Theory*, vol. 48, pp. 580–598, Mar. 2002.

[162] T. Can, N. Rengaswamy, R. Calderbank, and H. D. Pfister, "Kerdock codes determine unitary 2-designs," in *Proc. 2019 IEEE Int. Symp. Inf. Theory (ISIT)*, (Paris, France), pp. 2908–2912, July 2019.

[163] C. J. Ballance, T. P. Harty, N. M. Linke, M. A. Sepiol, and D. M. Lucas, "High-fidelity quantum logic gates using trapped-ion hyperfine qubits," *Physical Review Letters*, vol. 117, pp. 1–6, Aug. 2016.

[164] S. Bravyi and A. Kitaev, "Universal quantum computation with ideal clifford gates and noisy ancillas," *Physical Review A*, vol. 71, Feb. 2005.

[165] K. Fujii, "Noise threshold of quantum supremacy," *arXiv preprint*, 2016.

[166] E. Knill, "Fault-tolerant postselected quantum computation: Schemes," *arXiv preprint*, 2004.

[167] E. Knill, "Fault-tolerant postselected quantum computation: Threshold analysis," *arXiv preprint*, 2004.

[168] N. M. Linke, M. Gutierrez, K. A. Landsman, C. Figgatt, S. Debnath, K. R. Brown, and C. Monroe, "Fault-tolerant quantum error detection," *Science advances*, vol. 3, Oct. 2017.

[169] C. Chamberland, A. Kubica, T. J. Yoder, and G. Zhu, "Triangular color codes on trivalent graphs with flag qubits," *New Journal of Physics*, vol. 22, p. 023019, Feb. 2020.

[170] X. Xu, J. Sun, S. Endo, Y. Li, S. Benjamin, and X. Yuan, "Variational algorithms for linear algebra," *Science Bulletin*, vol. 66, pp. 2181–2188, Nov. 2021.

[171] G. E. Crooks, "Performance of the quantum approximate optimization algorithm on the maximum cut problem," *arXiv preprint*, 2018.

[172] J. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nat. Commun.*, vol. 9, pp. 1–6, Nov. 2018.

[173] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. J. Coles, "Noise-induced barren plateaus in variational quantum algorithms," *Nature communications*, vol. 12, pp. 1–11, Nov. 2021.

[174] R. Takagi, "Optimal resource cost for error mitigation," *Physical Review Research*, vol. 3, p. 033178, Aug. 2021.

[175] N. Moll, P. Barkoutsos, L. S. Bishop, J. M. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn, *et al.*, "Quantum optimization using variational algorithms on near-term quantum devices," *Quantum Science and Technology*, vol. 3, no. 3, 2018.

[176] P. J. J. O'Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, *et al.*, "Scalable quantum simulation of molecular energies," *Physical Review X*, vol. 6, no. 3, 2016.

[177] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, "Cost function dependent barren plateaus in shallow parametrized quantum circuits," *Nature communications*, vol. 12, pp. 1–12, Mar. 2021.

[178] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, "Cost-function-dependent barren plateaus in shallow quantum neural networks," *arXiv preprint*, 2020.

[179] G. Verdon, M. Broughton, J. McClean, K. J. Sung, R. Babbush, Z. Jiang, H. Neven, and M. Mohseni, "Learning to learn with quantum neural networks via classical neural networks," *arXiv preprint*, 2019.

[180] A. Skolik, J. McClean, M. Mohseni, P. van der Smagt, and M. Leib, "Layerwise learning for quantum neural networks," *Quantum Machine Intelligence*, vol. 3, pp. 1–11, Jan. 2021.

[181] J. I. Colless, V. V. Ramasesh, D. Dahlen, M. S. Blok, M. E. Kimchi-Schwartz, J. McClean, J. Carter, W. A. de Jong, and I. Siddiqi, "Computation of molecular spectra on a quantum processor with an error-resilient algorithm," *Physical Review X*, vol. 8, Feb. 2018.

[182] X. Bonet-Monroig, R. Sagastizabal, M. Singh, and T. E. O'Brien, "Low-cost error mitigation by symmetry verification," *Physical Review A*, vol. 98, Dec. 2018.

[183] J. M. Chow, J. M. Gambetta, A. D. Córcoles, S. T. Merkel, J. A. Smolin, C. Rigetti, S. Poletto, G. A. Keefe, M. B. Rothwell, J. R. Rozen, M. B. Ketchen, and M. Steffen, "Universal quantum gate set approaching fault-tolerant thresholds with superconducting qubits," *Physical Review Letters*, vol. 109, pp. 1–5, Aug. 2012.

[184] M. A. Cirone, A. Delgado, D. G. Fischer, M. Freyberger, H. Mack, and M. Mussinger, "Estimation of quantum channels with finite resources," *Quantum Information Processing*, vol. 1, pp. 303–326, Oct. 2002.

[185] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, "A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem," *Science*, vol. 292, no. 5516, pp. 472–475, 2001.

[186] S. Verdu *et al.*, *Multiuser detection*. Cambridge university press, 1998.

[187] P. Botsinis, S. X. Ng, and L. Hanzo, "Fixed-complexity quantum-assisted multi-user detection for CDMA and SDMA," *IEEE Transactions on Communications*, vol. 62, pp. 990–1000, Mar. 2014.

[188] H. Buhrman, R. Cleve, J. Watrous, and R. De Wolf, "Quantum fingerprinting," *Physical Review Letters*, vol. 87, p. 167902, Sept. 2001.

[189] A. K. Ekert, C. M. Alves, D. K. L. Oi, M. Horodecki, P. Horodecki, and L. C. Kwek, "Direct estimations of linear and nonlinear functionals of a quantum state," *Physical Review Letters*, vol. 88, p. 217901, May 2002.

[190] M. Wildemeersch, T. Q. S. Quek, M. Kountouris, A. Rabbachin, and C. H. Slump, "Successive interference cancellation in heterogeneous networks," *IEEE Transactions on Communications*, vol. 62, pp. 4440–4453, Dec. 2014.

[191] C.-F. Liu, M. Bennis, M. Debbah, and H. V. Poor, "Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing," *IEEE Transactions on Communications*, vol. 67, pp. 4132–4150, June 2019.

[192] M. A. Hanson, "Invexity and the Kuhn-Tucker theorem," *Journal of Mathematical Analysis and Applications*, vol. 236, pp. 594–604, Aug. 1999.

[193] J. J. Moré and D. J. Thuente, "Line search algorithms with guaranteed sufficient decrease," *ACM Trans. Math. Software (TOMS)*, vol. 20, pp. 286–307, Sept. 1994.

[194] F. Roosta, Y. Liu, P. Xu, and M. W. Mahoney, "Newton-MR: Newton's method without smoothness or convexity," *arXiv preprint*, 2018.

[195] S. Boyd, , and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[196] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," *Nature Physics*, vol. 10, pp. 631–633, July 2014.

[197] S. Salek, D. Ebler, and G. Chiribella, "Quantum communication in a superposition of causal orders," *arXiv preprint*, 2018.

[198] G. Chiribella and H. Kristjánsson, "Quantum Shannon theory with superpositions of trajectories," *Proc. Roy. Soc. A*, vol. 475, no. 2225, p. 20180903, 2019.

[199] P. A. Guérin, G. Rubino, and Č. Brukner, "Communication through quantum-controlled noise," *Physical Review A*, vol. 99, p. 062317, June 2019.

[200] G. Chiribella, M. Banik, S. S. Bhattacharya, T. Guha, M. Alimuddin, A. Roy, S. Saha, S. Agrawal, and G. Kar, "Indefinite causal order enables perfect quantum communication with zero capacity channels," *New Journal of Physics*, vol. 23, p. 033039, Mar. 2021.

[201] G. Brassard, P. Høyer, and A. Tapp, "Quantum counting," in *Proc. 25th International Colloquium on Automata, Languages, and Programming*, (Aalborg, Denmark), pp. 820–831, July 1998.

[202] M. Dobšíček, G. Johansson, V. Shumeiko, and G. Wendin, "Arbitrary accuracy iterative quantum phase estimation algorithm using a single ancillary qubit: A two-qubit benchmark," *Physical Review A*, vol. 76, p. 030306, Sept. 2007.

[203] N. Wiebe and C. Granade, "Efficient Bayesian phase estimation," *Physical Review Letters*, vol. 117, p. 010503, June 2016.

[204] A. D. Córcoles, M. Takita, K. Inoue, S. Lekuch, Z. K. Minev, J. M. Chow, and J. M. Gambetta, "Exploiting dynamic quantum circuits in a quantum algorithm with superconducting qubits," *Physical Review Letters*, vol. 127, p. 100501, Sept. 2021.

[205] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. USA: Prentice Hall Press, 3rd ed., 2009.

[206] F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proc. IEEE*, vol. 66, pp. 51–83, Jan. 1978.

[207] W. van Dam, G. M. D'Ariano, A. Ekert, C. Macchiavello, and M. Mosca, "Optimal quantum circuits for general phase estimation," *Physical Review Letters*, vol. 98, p. 090501, Mar. 2007.

[208] G. Rendon, T. Izubuchi, and Y. Kikuchi, "Effects of cosine tapering window on quantum phase estimation," *arXiv preprint*, 2021.

[209] S. Ye, J. Sun, and E. Aboutanios, "On the estimation of the parameters of a real sinusoid in noise," *IEEE Signal Process. Lett.*, vol. 24, pp. 638–642, May 2017.

[210] S. Djukanović, "An accurate method for frequency estimation of a real sinusoid," *IEEE Signal Process. Lett.*, vol. 23, pp. 915–918, July 2016.

[211] H. Shu, J. Wu, L. Senhadji, and L. Luo, "New fast algorithm for modulated complex lapped transform with sine windowing function," *IEEE Signal Process. Lett.*, vol. 16, pp. 93–96, Feb. 2009.

[212] S. M. Kay, *Fundamentals of statistical signal processing: Estimation theory*. Prentice-Hall, Inc., 1st ed., 1993.

[213] D. Wang, O. Higgott, and S. Brierley, "Accelerated variational quantum eigensolver," *Physical Review Letters*, vol. 122, p. 140504, Apr. 2019.

[214] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge University Press, 2nd ed., 2012.

[215] D. Pastor and Q.-T. Nguyen, "Random distortion testing and optimality of thresholding tests," *IEEE Transactions on Signal Processing*, vol. 61, no. 16, pp. 4161–4171, 2013.

[216] M. Van Hoeij, "Factoring polynomials and the knapsack problem," *Journal of Number Theory*, vol. 95, pp. 167–189, Aug. 2002.

[217] M. A. Hanson, "On sufficiency of the Kuhn-Tucker conditions," *Journal of Mathematical Analysis and Applications*, vol. 80, pp. 545–550, Apr. 1981.

[218] M. Syed, P. Pardalos, and J. Principe, "Invexity of the minimum error entropy criterion," *IEEE Signal Process. Lett.*, vol. 20, pp. 1159–1162, Dec. 2013.

[219] I. S. Gradshteyn and I. M. Ryzhik, *Table of integrals, series, and products*. Academic press, 2014.

# Subject Index

# Author Index