Constructing and Benchmarking Noise Models for Quantum Computing

Dissertation

zur Erlangung des Doktorgrades
an der Fakultät für Mathematik,
Informatik und Naturwissenschaften
der Universität Hamburg

vorgelegt beim Fach-Promotionsausschuss Informatik von

Tom Weber

Hamburg
2024

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Sofern im Zuge der Erstellung der vorliegenden Dissertationsschrift generative Künstliche Intelligenz (gKI) basierte elektronische Hilfsmittel verwendet wurden, versichere ich, dass meine eigene Leistung im Vordergrund stand und dass eine vollständige Dokumentation aller verwendeten Hilfsmittel gemäß der Guten wissenschaftlichen Praxis vorliegt. Ich trage die Verantwortung für eventuell durch die gKI generierte fehlerhafte oder verzerrte Inhalte, fehlerhafte Referenzen, Verstöße gegen das Datenschutz- und Urheberrecht oder Plagiate.

Tom Weber

Hamburg, 26. September 2024 Tom Weber

Gutachter:innen der Dissertation:	Prof. Dr. Kerstin Borras Prof. Dr. Ludwig Mathey Prof. Dr.-Ing. Ina Schäfer
Zusammensetzung der Prüfungskommission:	Prof. Dr. Kerstin Borras Prof. Dr. Ludwig Mathey Prof. Dr.-Ing. Ina Schäfer Prof. Dr.-Ing. Görschwin Fey Prof. Dr. Walid Maleej Prof. Dr. Matthias Rarey
Vorsitzender der Prüfungskommission:	Prof. Dr. Matthias Rarey
Datum der Disputation:	19. Dezember 2024

To the memory of Matthias Riebisch and André van Hoorn.

ABSTRACT

Context: The main obstacle of quantum computing on its way to scalability is the noisy nature of current hardware. Various types of errors lead to incorrect computational results, and measures such as quantum error mitigation are needed to counteract these errors. Many error mitigation techniques require a thorough understanding of the noise and its impact on computations. Therefore, accurate noise models are indispensable to facilitate meaningful quantum computing in the near future. They also enable realistic simulations of quantum devices. Since these devices are currently limited and quantum computing is expensive, researchers typically rely on simulations for testing their algorithms.

Objective: In this thesis, we aim to construct a realistic noise model for quantum computing and optimise its model parameters. The model construction should build on the underlying physical processes, and the number of parameters should scale well with the system size. We also address the evaluation and comparison of different noise models by benchmarks using quantum circuits specific to real-world applications. Finally, we aim to develop useful graphical representations of noise models based on the existing quantum circuit model.

Method: We present a benchmarking framework for quantum computing noise models and evaluate it based on benchmarking quality attributes from the literature. Moreover, we construct a quantum computing noise model and develop a training procedure for its parameters. We analyse the parameters of the trained noise model and the impact of different types of errors on computations. We benchmark the noise model with the above approach and compare it to a noise model provided by IBM's software development kit Qiskit. Finally, we develop an extension of the quantum circuit model to represent noise channels.

Result: The benchmarks performed in this thesis show that our noise model predicts noisy hardware behaviour of IBM's ibmq_manila quantum device equally well as the Qiskit model, if not better, based on quantum circuits commonly used for variational quantum algorithms. Within the trained noise model, readout error has the most detrimental impact on computations. Our benchmarking approach satisfies relevant quality criteria by choosing suitable quantum circuits and objective functions to compare model predictions to hardware data. Moreover, the quantum circuit model can be extended meaningfully by noise channels.

Conclusion: Our benchmarking approach is suitable for evaluating and comparing quantum computing noise models. In future work, volumetric benchmarks should be performed for various noise models and application contexts. The parameter optimisation based on learning with training circuits yields accurate noise models and shows the importance of mitigating measurement error. Different optimisation algorithms and quantum circuits should be explored, and more types of noise should be incorporated into the noise model.

ZUSAMMENFASSUNG

Kontext: Das größte Hindernis von Quantencomputing auf dem Weg zur Skalierbarkeit ist die fehlerhafte Natur aktueller Hardware. Verschiedene Fehlertypen führen zu inkorrekten Ergebnissen von Berechnungen, weshalb Gegenmaßnahmen wie Quantum Error Mitigation benötigt werden. Viele dieser Techniken setzen ein umfassendes Verständnis der Fehler und ihres Einflusses auf Berechnungen voraus. Genaue Fehlermodelle sind daher unverzichtbar, um sinnvolles Quantencomputing in naher Zukunft zu ermöglichen. Weiterhin erlauben sie realistische Simulationen von Quantencomputern. Da deren Zahl begrenzt und ihre Benutzung teuer ist, stützt sich die Forschung zu Testzwecken oft auf solche Simulationen.

Ziel: Diese Arbeit beschäftigt sich mit der Konstruktion realistischer Fehlermodelle für Quantencomputer und der Optimierung ihrer Parameter. Die Konstruktion sollte auf den zugrundeliegenden physikalischen Prozessen basieren und die Anzahl der Paremeter sollte gut mit der Systemgröße skalieren. Weiterhin befassen wir uns mit der Evaluation und dem Vergleich verschiedener Fehlermodelle durch Benchmarks anhand von anwendungsspezifischen Quantencircuits. Zuletzt setzen wir uns als Ziel, grafische Darstellungen von Fehlermodellen zu entwickeln.

Methodik: Wir entwickeln ein System zum Benchmarken von Fehlermodellen und bewerten dieses anhand von Qualitätskriterien für Benchmarks aus der Literatur. Außerdem konstruieren wir ein Fehlermodell und erarbeiten eine Methode zur Optimierung seiner Parameter. Weiterhin analysieren wir das trainierte Modell hinsichtlich seiner Parameter und der Auswirkungen verschiedener Fehler auf Berechnungen. Zur Evaluation führen wir die Benchmarks von oben durch und vergleichen unser Fehlermodell mit dem von IBMs Qiskit zur Verfügung gestellten Modell. Zuletzt erarbeiten wir eine Erweiterung des Quantencircuitmodells um Quantenfehler.

Ergebnisse: Die in dieser Arbeit durchgeführten Benchmarks zeigen, dass unser Fehlermodell das fehlerhafte Verhalten des ibmq_manila-Quantencomputers mindestens genauso gut, wenn nicht besser, beschreibt als das Modell aus Qiskit. Dabei nutzen wir Quantencircuits, die häufig für variationelle Quantenalgorithmen verwendet werden. Im trainierten Fehlermodell haben Messfehler den größten Einfluss auf Berechnungen. Unser Ansatz zum Benchmarking kann die Qualitätskriterien durch die Wahl geeigneter Quantencircuits und Funktionen zum Vergleich von Modellvorhersage und Hardwaredaten erfüllen. Weiterhin kann das Quantencircuitmodell sinnvoll um Fehlerprozesse erweitert werden.

Schlussfolgerung: Unser Benchmarking-Ansatz ist geeignet, um Fehlermodelle zu bewerten und miteinander zu vergleichen. In Zukunft sollten solche Benchmarks für verschiedene Modelle aus der Literatur und hinsichtlich unterschiedlicher Anwendungsgebiete durchgeführt werden. Die Optimierung der Modellparameter auf Grundlage vom Lernen auf Trainingscircuits führt zu realistischen Modellen und zeigt, wie wichtig die Milderung von Messfehlern ist. Andere Optimierungsalgorithmen und Quantencircuits sollten zukünftig hierfür erkundet und weitere Fehlertypen im Modell eingebaut werden.

ACKNOWLEDGEMENT

There is a long list of people who have supported me over the past four years, without whom this work would never have been possible. I will try my best to express my gratitude to everybody who has helped me in one way or another and to not forget anybody.

First and foremost, I want to thank Matthias Riebisch and André van Hoorn. With you, the world has not only lost two fantastic people but also two great supervisors. But since I know that none of you would have wanted any sorrow to find a place here, I will stick with the bright side of life.

Matthias, you have been my supervisor for almost three years. Thank you for your continuous support. Although you never really found your way with quantum mechanics, your advice has been indispensable. You have always encouraged me to follow my own ideas while ensuring that I progress nevertheless. And, perhaps most importantly, you have shown me the importance of not taking work too seriously. Finally, thank you for the annual delivery of wine and chocolate.

André, your unwavering support has been a cornerstone for our group. Your relentlessly optimistic attitude has been a driving force for us all. You have stepped in to supervise me without hesitation, and your guidance has been instrumental in helping me finish my thesis. I enjoyed every beer we shared.

I am grateful to my co-supervisors Kerstin Borras, Karl Jansen, and Dirk Krücker for their tireless guidance and our insightful discussions over the years. You have always brought me together with other researchers, constantly expanding my horizons. I want to thank you all for giving me the opportunity to become a doctoral student as part of this great project.

Furthermore, I want to thank Karl Jansen and Dirk Krücker for their helpful comment in my manuscript.

I am also grateful to everyone at SWK and DASHH for creating such an enjoyable working environment in Hamburg. Because of you, I have learned much more than just about quantum computers, and I have always had someone to talk to, which must not be underestimated in the life of a doctoral student.

Moreover, I want to thank my collaborators of the "Quantum Call" worldwide for countless discussions and our joint project on error mitigation. Although things did not go as expected, I have learned a lot during that time.

Finally, special thanks go to my friends and my family for their neverending support. To my parents, who have made it possible for me to pursue my path as a doctoral student. To my brother Leon, who will always find the silver lining in everything I tell him. To Karl and Tobi, who share similar experiences as PhD students and to whom I owe countless stupid laughter.

And last but most certainly not least, to Luisa, who always kept building me up when I struggled. Who made the past four years as a doctoral student the best time of my life. Thank you for all your love. And, of course, I must not forget Marley, who has been and will always be the best company I could have wished for.

This work was supported by DASHH (Data Science in Hamburg - Helmholtz Graduate School for the Structure of Matter) with the Grant-No. HIDSS-0002. I also acknowledge the use of IBM Quantum services for this work. The views expressed are those of the author, and do not reflect the official policy or position of IBM or the IBM Quantum team.

PUBLICATION LIST

I contributed to the following publications during my doctoral studies, where the contributions of all authors are described using the Contributor Role Taxonomy (CRediT) [1].

1. Tom Weber, Matthias Riebisch, Kerstin Borras, Karl Jansen, and Dirk Krücker [Mar. 2021]. "Modelling for Quantum Error Mitigation". *2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C)*. 2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C), pp. 102–105. DOI: 10.1109/ICSA-C52384.2021.00026

   Author contributions: I was responsible for everything regarding conceptualisation, methodology, visualisation, and writing the original draft. K.B., K.J., D.K., and M.R. have contributed in discussions on conceptualisation and by reviewing the manuscript.

2. Constantia Alexandrou, Lena Funcke, Tobias Hartung, Karl Jansen, Stefan Kühn, Georgios Polykratis, Paolo Stornati, Xiaoyang Wang, and Tom Weber [July 8, 2022]. "Investigating the variance increase of readout error mitigation through classical bit-flip correction on IBM and Rigetti quantum computers". *Proceedings of The 38th International Symposium on Lattice Field Theory — PoS(LATTICE2021)*. Vol. 396. Conference Name: The 38th International Symposium on Lattice Field Theory. SISSA Medialab, p. 243. DOI: 10.22323/1.396.0243

   Author contributions: This publication serves more as a motivation for the work presented in this thesis and is not directly related. I was reponsible for conceptualisation, formal analysis, software, methodology, and editing and reviewing the manuscript. In particular, I wrote the code for the experiments on the Rigetti machine, I ran these experiments, and I evaluated the resulting data. Moreover, I wrote part of the original draft and edited the manuscript. C.A., L.F., K.J., and X.W. were responsible for conceptualisation, methodology, and editing and reviewing the manuscript. T.H.,

[1] See https://credit.niso.org/

S.K. and P.S. were responsible for conceptualisation, formal analysis, methodology, and editing and reviewing the manuscript. G.P. was responsible for conceptualisation, formal analysis, methodology, software, visualisation, and writing the original draft.

3. Tom Weber, Kerstin Borras, Karl Jansen, Dirk Krücker, and Matthias Riebisch [May 2024]. "Construction and volumetric benchmarking of quantum computing noise models". *Physica Scripta* 99.6. Publisher: IOP Publishing, p. 065106. DOI: 10.1088/1402-4896/ad406c

   Author contributions: I was responsible for everything concerning conceptualisation, formal analysis, methodology, software, visualisation, and writing the original draft. More concretely, I translated the volumetric benchmarking framework to quantum computing noise models and evaluated the benchmarks using well-known criteria from the literature. Moreover, I wrote the code to generate and run all quantum circuits on the IBM device and simulate them on classical hardware. I trained all noise models and evaluated the resulting data. K.B., K.J., D.K., and M.R. have contributed in discussions on conceptualisation, methodology, and by reviewing the manuscript. Furthermore, D.K. has helped with the formal analysis, patricularly for generating the confidence intervals.

CONTENTS

CHAPTER

ONE

INTRODUCTION

Quantum computers are expected by many to surpass their classical counterparts in terms of computing time, memory usage, energy consumption, or accuracy for specific applications in the future. Whether they will ever fully hold up to these expectations is an open question that will not be discussed here. While the first visions of a quantum computer had already emerged in the 1980s [Feynman, 1982], recent developments have revived these visions by realising constantly improved quantum devices. This steady progress of quantum hardware has led to the first claims of quantum advantage, where quantum computers could solve certain artificial problems faster than classical devices [Arute et al., 2019; Zhong et al., 2020; Wu et al., 2021].

However, quantum computers are prone to noise, limiting their use for practical applications and hindering them on their way to scalability. The noisy hardware makes reliable computations nearly impossible and raises the need for countermeasures. Quantum error correction is a potential future solution but is prohibited by the size of current machines as it requires a significant overhead of qubit resources [Devitt et al., 2013]. Several publications propose alternative approaches to reduce the impact of noise on computational results without additional qubits [Li and Benjamin, 2017; Temme et al., 2017]. These approaches are called quantum error mitigation, and they aim to estimate correct computational results based on erroneous ones, exploiting knowledge about the effect of noise on the computation [Cai et al., 2023].

Recent research has proven that error mitigation can pave the way for reliable quantum computing with Noisy Intermediate-Scale Quantum (NISQ) devices. Kim, Wood, et al., 2023 have used quantum error mitigation to simulate physical systems and arguably obtain the first useful results from a quantum computer for an interesting problem. While the specific mitigation protocol they have used - zero-noise extrapolation - does not require an accurate understanding of the errors (but a higher degree of system control), many other protocols rely on detailed knowledge about the noise. Moreover, all quantum error mitigation techniques cause a computational overhead to compensate for a higher variance of the results [Takagi et al., 2022]. Combining many mitigation methods is typically not feasible, and prioritising the most dominant errors is necessary. This prioritisation again requires an accurate characterisation of quantum noise.

Noise models provide a means of mathematically describing quantum noise. They can be used to simulate quantum circuits and predict their erroneous outcomes, offering an alternative to using expensive quantum hardware for testing purposes. A noise model consists of errors affecting different parts of the computation and can help us understand the noisy processes in quantum computing. Realistic noise models are essential to improving quantum error mitigation and quantum computing. In this thesis, we face several challenges related to noise models.

1.1. Motivation and Problem Statement

As explained above, accurate noise models for quantum computing are essential to implementing quantum error mitigation successfully. Their importance gets amplified because they can provide realistic simulations of quantum computers. Since quantum computers are limited and their use is expensive, it is often feasible to use simulators instead, and better noise models improve the ability of these simulators to emulate real quantum devices. We address different aspects of noise models and discuss their respective challenges in the following.

Our primary focus is their *construction*. Building accurate noise models and optimising their parameters is the main goal of this thesis. But how can we measure the accuracy of noise models? To answer this question is our next central objective. The systematic *evaluation and comparison* of noise models yield valuable information for choosing error mitigation techniques. Finally, we aim to enhance our understanding of quantum noise by a suitable *visualisation* of noise models, graphically representing the errors and the computational operations that are affected.

Noise model construction: A single quantum error affecting N qubits can generally be represented by a $4^N \times 4^N$ matrix, i.e., by exponentially many parameters. Much work has been published addressing the characterisation of quantum noise in terms of such matrices, for example, tomography methods such as quantum process tomography [Chuang and Nielsen, 1997], gate set tomography [Merkel et al., 2013] or randomised benchmarking [Emerson et al., 2007]. However, they do not give us insight into the underlying physical processes that lead to the noise. This insight can provide valuable information about the behaviour of a quantum computer, for example, if thermal relaxation and dephasing play a significant role in noise. We can counteract specific noise using targeted error mitigation and, in the future, also error correction.

Noise models based on physical processes can therefore be beneficial, but not many such noise models have been discussed in the literature. IBM's software development kit Qiskit [Qiskit contributors, 2023] provides noise models for each of their quantum devices. They consider measurement erroneous and quantum gates affected by depolarisation, thermal relaxation, and dephasing. Georgopoulos et al., 2021a extend this model by adding state preparation errors. Neither model considers crosstalk error, which describes interactions amongst qubits and has been found in current quantum hardware [Rudinger et al., 2021]. Constructing more advanced noise models that include crosstalk error is the natural next step for a better characterisation of quantum noise. Since noise models depend on certain parameters such as error probabilities, reliable methods to obtain suitable parameters are crucial.

Benchmarks for noise models: The number of approaches to characterising quantum noise is constantly growing, and it is becoming increasingly difficult to assess which one predicts noisy hardware behaviour best. To our knowledge, no systematic benchmark approach exists for evaluating and comparing different noise models. Publications on noise models typically use a few simple quantum circuits that are run on a quantum computer and simulated with a noise model to compare the results and determine the deviation between model predictions and reality. Moreover, the used quantum circuits are often shallow with constant depth, and dependencies of the quality of a noise model on the problem scale are not taken into account.

A systematic benchmark with quantum circuits motivated by real-world applications of varying sizes could reveal interesting properties of different noise models. For example, a noise model might be capable of describing measurement error accurately but not gate errors. Such a noise model could be useful for simulating shallow quantum circuits with only a few gates. The more gates a quantum circuit has, the more dominant gate errors become, and the model predictions deteriorate. Since noise models could also be tailored for specific applications, the quantum circuits to test them should be similar to practical applications.

Model visualisation: Accurate noise models can benefit not only quantum error mitigation and realistic quantum simulations but also our sense of noisy processes in a quantum computer. While the mathematical descriptions of quantum noise are the most complete, graphical representations could improve our understanding and communication regarding noise models. Several publications have produced figures that contain quantum circuits with different symbols depicting errors, for example [Wallman and Emerson, 2016]. However, quantum errors are fundamentally different operations than quantum gates, and the interpretation of a quantum circuit must be changed upon adding noise channels. A rigorous treatment of extending the quantum circuit model does not yet exist.

1.2. Overview of Contribution

After highlighting the significance of accurate noise models for quantum computing, we briefly present our research approach. To address the different challenges of understanding, evaluating, and constructing such noise models, we formulate three research questions that are further discussed later.

> **RQ1**: How can noise models and their impact on quantum circuits be visualised?
>
> **RQ2**: How can noise models be evaluated and compared regarding their accuracy?
>
> **RQ3**: How can realistic noise models be constructed?

Each research question is addressed in one of the main chapters of this thesis, which we summarise in the following.

Chapter 5 - We propose extending the quantum circuit model by noise channels to visualise noise models. Quantum circuits are the prevalent way to model quantum computations and provide an intuitive graphical representation thereof. However, the quantum circuit model only describes qubit initialisation, unitary quantum gates, and measurement. Quantum noise is a more general process than the above operations. In our approach, noise channels are depicted by rounded boxes in a quantum circuit, and another box indicates which operation is affected. We provide a rigorous definition of noisy quantum circuits in terms of directed graphs.

Chapter 6 - We present a volumetric benchmarking framework for evaluating and comparing noise models. Initially introduced to assess the fidelity of quantum hardware by Blume-Kohout and Young, 2020, we translate the concept to test noise models for their accuracy. The benchmarks compare the predictions of a noise model to the results from quantum hardware for a collection of test quantum circuits. These quantum circuits are representative of a quantum computing application of interest and vary in size, hence the name volumetric benchmarks. In the following, we briefly discuss the benchmark procedure. ① For different circuit sizes, define a set of quantum circuits. ② Set up compilation rules to transform the quantum circuits to native gates of the quantum device. ③ Determine a way to obtain the model predictions by noisy simulations of the quantum circuits. ④ Run the quantum circuits on the quantum computer. ⑤ Compare model predictions to hardware data for a single circuit. ⑥ Compute an overall evaluation based on the individual circuits. We evaluate this approach regarding established benchmark quality attributes. Moreover, we discuss possibilities to ensure the quality by appropriate choices of quantum circuits and objective functions to compare model predictions to hardware data.

Chapter 7 - We present a noise model including the following five types of noise:

- **State preparation error:** The preparation of an initial qubit state is erroneous, resulting in a quantum bit-flip with a certain probability. This probability can differ from qubit to qubit.
- **Measurement error:** Measurements at the end of a quantum circuit yield incorrect results. We model this error by a classical bit-flip occurring after the measurement, and the probabilities can be asymmetric regarding the direction of the flip.
- **Depolarising error:** Quantum gates are affected by depolarisation, occasionally resulting in a completely mixed qubit state. The depolarising parameters can vary for all gates on all qubits (or qubit pairs in the case of two-qubit gates).
- **Thermal relaxation and dephasing error:** Thermal relaxation is the transition of qubits towards their ground state over time, while dephasing drives them to a classical state. Dephasing and relaxation depend on the duration of quantum computations, which is mainly caused by the execution of quantum gates.
- **Crosstalk error:** Crosstalk error describes unwanted interactions between qubits. Here, we consider the situation where applying a rotation gate to a qubit causes rotations around the same axis but with possibly different angles on neighbouring qubits.

The above noise model depends on various parameters, which must be determined accurately to make the model as realistic as possible. We propose a machine learning-like approach to optimise the parameters. For a set of training quantum circuits, we repeatedly compare noise model predictions to hardware results and adapt the parameters accordingly. Different optimisation algorithms and loss functions are discussed. Moreover, the resulting model parameters and the impact of different error types on computational results are analysed.

While there are various quantum computing paradigms, such as measurement-based [Raussendorf and Briegel, 2001; Nielsen, 2003], annealing [Kadowaki and Nishimori, 1998], topological [Kitaev, 2003], or adiabatic [Farhi, Goldstone, Gutmann, et al., 2001], this thesis only considers gate-based universal quantum computing (for an overview, see [Nielsen and Chuang, 2010]). The noise model in Chapter 7 is constructed using superconducting qubits provided by IBM. However, the method is independent of the hardware realisation and can easily be used to build noise models describing alternative approaches, for example, based on neutral atoms or ion traps.

1.3. Thesis Structure

This thesis is structured as follows. Chapter 2 presents the necessary foundations of quantum mechanics. It comprehensively introduces quantum computing, quantum noise, and benchmarking. Chapter 3 discusses related work, and Chapter 4 gives an overview of our research. Chapters 5, 6, and 7 present our main contributions, as explained above. Finally, Chapter 8 concludes this thesis by summarising the results and discussing future work.

CHAPTER

TWO

FOUNDATIONS

2.1. Quantum Computing

This section summarises the fundamental quantum computing concepts needed to understand quantum errors and noise models. It introduces qubits in Section 2.1.1, quantum gates in Section 2.1.2, and the quantum circuit model in Section 2.1.3. Finally, it explains how such circuits can be run on current quantum hardware or simulators in Section 2.1.4. All sections focus on gate-based quantum computing and do not consider other computing paradigms. This chapter mainly follows [Nielsen and Chuang, 2010], where further discussions about the subjects can be found.

2.1.1. Qubits

A classical computer uses bits as fundamental units of information, where a bit can take one of precisely two states called 0 and 1. In quantum computing, bits are replaced by their quantum counterpart: qubits. They exploit quantum-mechanical principles such as superposition and entanglement, offering the potential to overcome different challenges of classical computation related to memory, time efficiency, or accuracy [Arute et al., 2019; Daley et al., 2022; Zhong et al., 2020]. A prominent example is Shor's algorithm for efficiently finding prime factors of integer numbers [Shor, 1997].

Mathematically, a qubit is a two-dimensional quantum system that can be described by a Hilbert space $\mathcal{H} \simeq \mathbb{C}^2$ with a computational basis consisting of two states $|0\rangle, |1\rangle \in \mathcal{H}$. Here, as well as throughout this thesis, we use the Dirac notation to denote quantum states. More details on this notation can be found in Appendix A or in [Nielsen and Chuang, 2010]. While a bit can only be in either of two states, a qubit can also be in a superposition

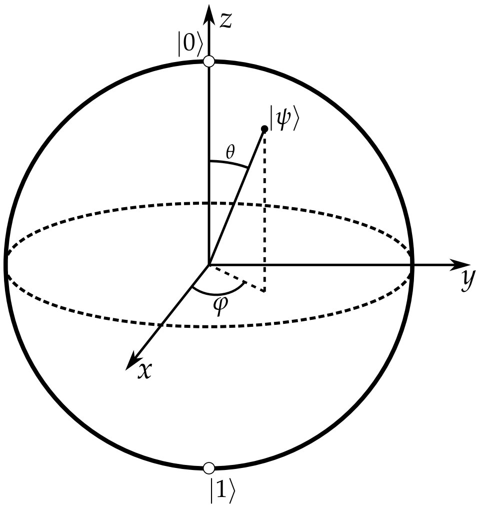


Figure 2.1.: The Bloch sphere graphically represents a single qubit.

of the basis states. In general, it can be expressed as a vector

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

where $\alpha, \beta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 = 1$. Since this state vector is two-dimensional, one can also identify the basis states with the column vectors

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \ . \tag{2.2}$$

With this identification, the general state $|\psi\rangle$ from (2.1) can be written as

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \ . \tag{2.3}$$

Besides the mathematical description of a qubit, the Bloch sphere shown in Figure 2.1 offers an instructive graphical representation. Its poles represent the two basis states and the general state of a qubit can be located anywhere on the sphere, while a classical bit would be restricted to the poles. Note that the state $|\psi\rangle$ in the figure is characterised by two angles θ, φ. In the notation from (2.1), this corresponds to the state

$$|\psi\rangle = \cos\frac{\theta}{2}\,|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}\,|1\rangle \ . \tag{2.4}$$

An essential property of a classical bit is the possibility to read out its current state. The situation is fundamentally different for a qubit. In general, getting complete information about the vector $|\psi\rangle$ describing its state is impossible [Nielsen and Chuang, 2010]. Instead, one can perform measurements of the qubit in the computational basis $\{|0\rangle, |1\rangle\}$. In such a measurement of $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, there are two possible outcomes: $|0\rangle$ or $|1\rangle$. They are obtained with the following probabilities:

$$p(0) = |\alpha|^2, \quad p(1) = |\beta|^2 . \tag{2.5}$$

Many measurements are needed to gain reliable information about these probabilities, but one cannot learn the true values of α and β including their phases. Moreover, the state of a qubit is generally changed upon measurement. After measuring a superposition $\alpha|0\rangle + \beta|1\rangle$ with outcome $|k\rangle$ ($k = 0, 1$), its state collapses to this measured state.

In addition to superposition, entanglement is another key concept of the quantum world that quantum computers aim to utilise. Entanglement is a phenomenon that can be observed for systems of multiple qubits and describes correlations between them.

A set of N qubits with corresponding spaces $\mathcal{H}_1, \ldots, \mathcal{H}_N$ can be represented by the tensor product $\mathcal{H} = \mathcal{H}_1 \otimes \ldots \otimes \mathcal{H}_N$. This space has dimension 2^N and its basis states are the tensor products

$$|k_1 \ldots k_N\rangle := |k_1\rangle \otimes \ldots \otimes |k_N\rangle \tag{2.6}$$

of the single-qubit basis states, where $k_i = 0, 1$ and $|k_i\rangle$ corresponds to the ith qubit. For example, the computational basis for two qubits consists of the four states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$. Therefore, an arbitrary two-qubit state can be written as a superposition

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \ , \tag{2.7}$$

where$|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$. Measuring this state in the computational basis yields the result $|k_1k_2\rangle$ with probability $\left|\alpha_{k_1k_2}\right|^2$. Everything can naturally be generalised to higher dimensions. Note that one can still identify the basis states of an N-qubit system with N-dimensional column vectors. In the example $N = 2$ from above, this leads to the vectors

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} . \tag{2.8}$$

To demonstrate entanglement, consider two qubits in the state

$$|\psi\rangle = \frac{1}{\sqrt{2}}(\,|00\rangle + |11\rangle\,) . \tag{2.9}$$

This state cannot be written as a single tensor product $|\psi_1\rangle \otimes |\psi_2\rangle$ with $|\psi_i\rangle \in \mathcal{H}_i$, as we show in the following. We can use (2.1) to write the states $|\psi_i\rangle$ in their most general form as $|\psi_i\rangle = \alpha_i |0\rangle + \beta_i |1\rangle$ and compute their tensor product:

$$\begin{aligned} |\psi_1\rangle \otimes |\psi_2\rangle &= [\alpha_1 |0\rangle + \beta_1 |1\rangle] \otimes [\alpha_2 |0\rangle + \beta_2 |1\rangle] \\ &= \alpha_1\alpha_2 |00\rangle + \alpha_1\beta_2 |01\rangle + \beta_1\alpha_2 |10\rangle + \beta_1\beta_2 |11\rangle \ . \end{aligned} \tag{2.10}$$

This can only yield the expression of (2.9) if $\alpha_1\alpha_2 = \beta_1\beta_2 = 1$ and $\alpha_1\beta_2 = \beta_1\alpha_2 = 0$, which is not possible simultaneously.

If we prepare two qubits as above, the overall system is in a well-defined state but the single qubits are not. Moreover, the only two possible measurement results are the states $|00\rangle$ and $|11\rangle$ with equal probabilities. In particular, measuring one qubit is already sufficient to know the measurement outcome for the second one. This correlation is called entanglement.

Superposition and entanglement are two fundamental aspects of quantum mechanics that are not accessible with classical computers. They enable completely new computations, potentially leading to advantages of quantum computers over their classical counterparts for certain applications.

2.1.2. Quantum Gates

The ability to control the state of classical bits is crucial to achieving meaningful classical computations. One way to describe changes of their state is given by logic gates. Logic gates acting on m bits and yielding n bits are maps $\{0,1\}^m \rightarrow \{0,1\}^n$. A basic example of a gate acting on a single bit is the so-called `NOT`-gate. It changes the state of a bit from 0 to 1 and vice versa. The `AND` gate is an example of a logic gate where the number of input bits differs from the number of outcome bits. It maps two bits to a single one and yields the result 1 if and only if both input bits are in the state 1.

Similar to bits, logic gates have their quantum counterpart. Quantum gates act on a set of qubits and change their state. Since qubits are two-dimensional quantum systems instead of binary values, the nature of quantum gates also differs fundamentally from logic gates. A quantum gate acting on N qubits described by the space $\mathcal{H}$ is a unitary $2^N \times 2^N$ matrix $U : \mathcal{H} \rightarrow \mathcal{H}$, i.e., a matrix with complex entries such that $U^\dagger = U^{-1}$, where $U^\dagger$ denotes the transpose matrix with complex conjugate entries. The unitarity of U ensures that the outcome of a quantum gate is a valid qubit state with measurement probabilities adding up to 1 again. Basic examples for quantum gates acting on a single qubit are the

Pauli matrices

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} . \tag{2.11}$$

With the column vector representations of the computational basis states from (2.3), one can compute:

$$\begin{aligned} X\left|0\right\rangle &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \left|1\right\rangle , \\ X\left|1\right\rangle &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \left|0\right\rangle . \end{aligned} \tag{2.12}$$

Therefore, the X gate maps $\left|0\right\rangle$ to $\left|1\right\rangle$ and vice versa and is the quantum analog to the `NOT` gate, also called the quantum bit-flip. Since all quantum gates are linear, the action of X on a general state can be computed as

$$X(\alpha\left|0\right\rangle + \beta\left|1\right\rangle) = \alpha\left|1\right\rangle + \beta\left|0\right\rangle . \tag{2.13}$$

Many other quantum gates needed in this thesis can be derived from the Pauli gates above. Firstly, consider the $\sqrt{X}$ gate

$$\sqrt{X} = \frac{1}{2} \begin{pmatrix} 1+\mathrm{i} & 1-\mathrm{i} \\ 1-\mathrm{i} & 1+\mathrm{i} \end{pmatrix} . \tag{2.14}$$

It satisfies $\sqrt{X} \cdot \sqrt{X} = X$ and is not particularly interesting at this point, but the $\sqrt{X}$ gate becomes important for practical purposes in later chapters. Secondly, given one of the Pauli matrices $P \in \{X, Y, Z\}$, one can define a new gate operation

$$R_p(\vartheta) := \exp\left(-\mathrm{i}\frac{\vartheta}{2}P\right) . \tag{2.15}$$

This new operation depends on a real parameter $\vartheta \in [0, 2\pi)$ and describes a rotation of the qubit around the p axis in the Bloch sphere representation. The explicit matrix forms of these rotation gates are

$$\begin{aligned} R_x(\vartheta) &= \begin{pmatrix} \cos\frac{\vartheta}{2} & -\mathrm{i}\sin\frac{\vartheta}{2} \\ -\mathrm{i}\sin\frac{\vartheta}{2} & \cos\frac{\vartheta}{2} \end{pmatrix} , \\ R_y(\vartheta) &= \begin{pmatrix} \cos\frac{\vartheta}{2} & -\sin\frac{\vartheta}{2} \\ \sin\frac{\vartheta}{2} & \cos\frac{\vartheta}{2} \end{pmatrix} , \\ R_z(\vartheta) &= \begin{pmatrix} e^{-\mathrm{i}\frac{\vartheta}{2}} & 0 \\ 0 & e^{\mathrm{i}\frac{\vartheta}{2}} \end{pmatrix} . \end{aligned} \tag{2.16}$$

So far, we have only considered gate operations acting on single qubits. However, quantum computing aims to exploit entanglement, which only occurs in situations with multiple qubits. Therefore, gates affecting multiple qubits are of significant importance. In this thesis, we only use one of these gates, the CX gate or *controlled* X gate. It acts on two qubits called control and target qubit. If the control qubit is in the state $|0\rangle$, the X gate from above is applied to the target qubit. Otherwise, the qubits are unchanged. Using the column vector representations of two-qubit basis states from (2.8), the CX gate can be written as

$$CX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} . \tag{2.17}$$

A basic computation shows that CX indeed leaves the basis states $|00\rangle$ and $|01\rangle$ invariant and acts as $CX\,|10\rangle = |11\rangle$, $CX\,|11\rangle = |10\rangle$. Moreover, we can easily observe that it can create entanglement. Consider the state $|\psi\rangle = 1/\sqrt{2}(|0\rangle + |1\rangle) \otimes |0\rangle$. This state is not entangled, but the action of CX on this state yields

$$CX\,|\psi\rangle = \frac{CX\,|00\rangle + CX\,|10\rangle}{\sqrt{2}} = \frac{|00\rangle + |11\rangle}{2} , \tag{2.18}$$

which is the entangled state from (2.9).

This concludes the introduction of specific quantum gates in this thesis. Further gates are not needed because of the following important result (a proof can be found in [DiVincenzo, 1995]):

Proposition 2.1.1. *Every quantum gate can be represented as a composition using only CX and single qubit gates.*

It turns out that the single qubit gates introduced in this section are already sufficient for building arbitrary quantum gates. A set of gates that can be used to produce every unitary

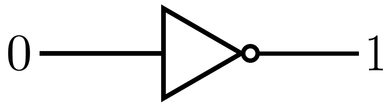


Figure 2.2.: Classical logic circuit with a single `NOT` gate.

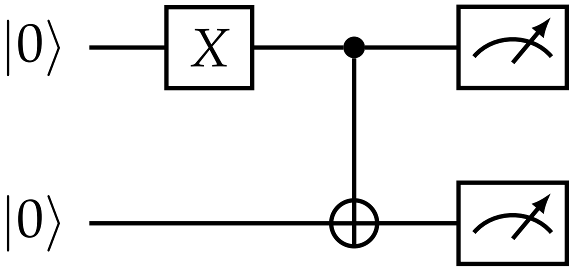


Figure 2.3.: Basic two qubit quantum circuit.

matrix on a set of qubits is called *universal gate set*. One possible choice for a universal gate set is

$$\mathcal{G} = \{X, \sqrt{X}, R_z, CX\}\,, \tag{2.19}$$

as we briefly explain in the following. Every single qubit gate can be built (up to a phase) from the rotation gates $R_x, R_y.R_z$ [Williams, 2011]. Moreover, one can compute that

$$R_y(\vartheta) = X\sqrt{X}R_z(\vartheta)\sqrt{X} \tag{2.20}$$

$$R_x(\vartheta) = R_y\left(\frac{\pi}{2}\right)R_z(\vartheta)R_y\left(-\frac{\pi}{2}\right)\,. \tag{2.21}$$

Proposition 2.1.1 and the equations above show that $\mathcal{G}$ is universal. Note that the X gate is not necessary because it can be represented by two consecutive $\sqrt{X}$ gates.

2.1.3. Quantum Circuits

Classical computations can be represented using logic circuits. They consist of preparing a set of bits in an initial state, acting on these bits with logic gates, and reading out the bits in the end. A logic circuit is a graphical model that uses wires to represent bits and different symbols representing the logic gates. Figure 2.2 shows an example of a basic logic circuit that only incorporates one bit and a single `NOT` gate.

The prevalent method to represent quantum computations is the quantum circuit model presented in [Deutsch, 1989]. Similarly to logic circuits, a quantum circuit uses wires and gates to describe the computational process. Figure 2.3 shows a quantum circuit acting on two qubits, which will serve as the running example throughout this thesis. In the beginning, both qubits are initialised in the $|0\rangle$ state. The state preparation is followed by an X gate acting on the first qubit and a CX gate with the first qubit as the control and the second qubit as the target qubit. Finally, both qubits are measured in the computational basis. These operations correspond to the basic building blocks of quantum circuits that we explain in the following.

- **State preparation:** Each qubit is prepared in its initial state at the beginning of a quantum circuit. The state $|0\rangle$ is typically chosen, as in the example above. This is also the case for all quantum circuits used later in this thesis. State preparation is graphically represented by writing the initial state next to the left end of a wire:

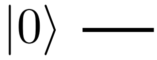


Figure 2.4.: State preparation.

- **Gate operations:** Quantum gates are applied to the qubits. There are various symbols representing different gates. They connect the incoming and outgoing wires of the affected qubits. The following figure shows the relevant quantum gates from Section 2.1.2.

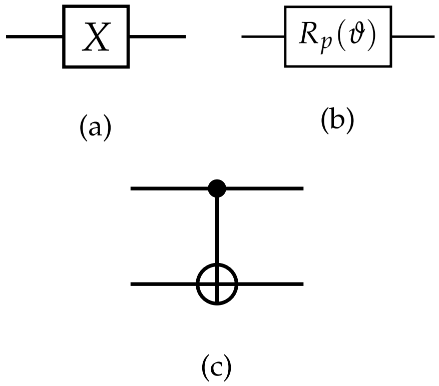


Figure 2.6.: (a) X gate, (b) Rotation gate, (c) CX gate.

- **Measurement:** Measurements in the computational basis can be performed at the end of a quantum circuit. In this thesis, quantum circuits always end with the measurements of all qubits. Measuring a computational basis state $|k_1 \ldots k_N\rangle$ yields a string of classical bits, or bitstring, '$k_1 \ldots k_N$'. For example, the bitstring '01' is obtained upon a measurement of the state $|01\rangle$. The measurement of a qubit is represented by a meter symbol next to the right end of a wire:

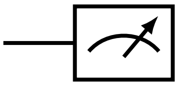


Figure 2.7.: Measurement in the computational basis.

All quantum circuits can be built from these components; see Figure 2.3 as an example. We can directly compute the ideal outcome of this specific circuit as follows. After the qubits are prepared in the state $|00\rangle$, the X gate changes this state to $|10\rangle$, followed by the CX gate that yields $|11\rangle$. The measurements should then always result in the bitstring '11'.

In Section 2.1.3, we explained that many measurements of a qubit in an arbitrary state $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ are needed to gain information about the probabilities $|\alpha|^2$ and $|\beta|^2$. Similarly, quantum circuits are usually run many times to obtain useful computational results. The number of runs is often referred to as the *number of shots* s.

For many quantum computing applications, such as Variational Quantum Eigensolver (VQE) [Peruzzo et al., 2014], it is feasible to create an arbitrary qubit state. A quantum circuit that begins with preparing the $|0\dots0\rangle$ state and is capable of this task is called fully expressive. Given a random quantum circuit, its expressivity is not obvious. On the one hand, it may not include enough gates to be fully expressive. On the other hand, there may be gates that do not contribute to the expressivity. In [Funcke, Hartung, Jansen, Kühn, and Stornati, 2021], the authors propose a method to evaluate the expressivity of quantum circuits and the usefulness of a single quantum gate for achieving it.

2.1.4. Quantum Hardware and Simulators

The physical realisation of a quantum computer is one of the main challenges of the NISQ era [Preskill, 2018]. Various requirements must be met by a potential quantum device to enable practical quantum computation by running quantum circuits. They directly correspond to the building blocks of quantum circuits discussed above. Following [DiVincenzo, 2000], these conditions are:

1. A physical system implementing a set of qubits.
2. The ability to prepare the qubits in a known initial state.
3. A universal gate set.
4. Qubit-specific measurements.
5. Much longer decoherence times than the time needed to apply a quantum gate.

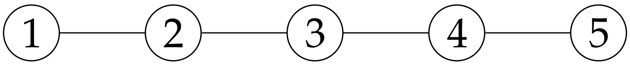


Figure 2.8.: Linear qubit layout.

While the requirements 1.-4. directly correspond to the building blocks of quantum circuits discussed above, the last one ensures the robustness of quantum information for the duration of quantum computations. To date, several approaches have emerged to develop a device that comes close to meeting these requirements. These approaches include superconducting qubits [Clarke and Wilhelm, 2008], ion traps [Cirac and Zoller, 1995], photonic devices [Knill, Laflamme, and Milburn, 2001], neutral atoms [Khazali and Mølmer, 2020; Henriet et al., 2020], and many others. All quantum experiments described in this thesis were performed on a superconducting device, and we do not discuss the physical realisations further.

Native gates

A quantum computer with a universal gate set does not necessarily offer a physical implementation for every quantum gate. Instead, quantum circuits need to be transformed into gates that the hardware can execute. These gates form the so-called *native gate set*, and the transformation process is called *transpiling*. While most machines only implement single and two-qubit gates, they typically still have different native gates. Moreover, the transpilation of a quantum circuit into native gates is generally not unique, but several compositions can yield the same computation. In the context of this thesis, it is crucial to keep track of the exact transpilation.

Qubit layouts

As explained in Section 2.1.2, two-qubit gates such as the CX gate can create entanglement. However, current quantum computers usually do not implement two-qubit gates between every pair of qubits. In other words, they do not offer full connectivity. The possible connections result in a specific qubit layout or qubit architecture. Figure 2.8 shows a possible linear layout, where five qubits are represented in a one-dimensional lattice, and only neighboured qubits are connected.

Simulators

Since quantum hardware is currently limited and, as discussed in Section 2.2, imperfect, it is often feasible to simulate quantum circuits. Various software packages offer such simulations, for example Pennylane, Qiskit, and QuEST [Bergholm et al., 2022; Qiskit contributors, 2023; Jones et al., 2019]. Since the state space of a set of qubits grows exponentially with their number, simulating large systems is very time-consuming, if not impossible.

In this thesis, we distinguish between two types of simulations. Firstly, quantum circuits can be simulated by generating measurement results in the form of shots, similar to real quantum computers. They represent their random behaviour, leading to the same challenges due to limited shots. Secondly, one can perform exact computations and directly predict a quantum circuit's true outcome probability distribution instead of an empirical one. Moreover, other variables of interest can sometimes be computed directly and do not need to be derived from a finite sample of runs.

2.1.5. Observables and Expectation Values

In quantum mechanics, an *observable* is a measurable physical quantity. If the system is described by a Hilbert space $\mathcal{H}$, then observables are Hermitian operators $O\colon \mathcal{H} \to \mathcal{H}$, i.e., linear operators with $O = O^\dagger$. The quantum mechanical *expectation value* of an observable O in a state $|\psi\rangle$ is defined as

$$\langle O \rangle := \langle \psi | O | \psi \rangle \ . \tag{2.22}$$

The Pauli matrices from (2.11) are important examples of observables because they can be directly measured on a quantum computer. If a quantum circuit prepares a single qubit in a state $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, then the expectation value of the Z operator is

$$\langle Z \rangle = \langle \psi | Z | \psi \rangle = |\alpha|^2 - |\beta|^2 \, . \tag{2.23}$$

Using a quantum device and running the circuit s times, this expectation value can be measured as

$$\langle Z \rangle = \frac{1}{s} \sum_{j=1}^{n} (-1)^{k(j)} \, , \tag{2.24}$$

where $k(j) \in \{0, 1\}$ denotes the measurement outcome of the jth run. Measuring the operators X and Y is achieved by adding corresponding gates to the quantum circuit. Moreover, one can construct more general N-qubit Pauli operators

$$\mathcal{P} = P_1 \otimes \ldots \otimes P_N \tag{2.25}$$

and linear combination thereof, where $P_i \in \{I, X, Y, Z\}$. The expectation value $\left\langle Z^{\otimes N} \right\rangle$ can be performed on a quantum computer by computing

$$\left\langle Z^{\otimes N} \right\rangle = \frac{1}{s} \sum_{j=1}^{n} (-1)^{k_1(j)} \cdot \ldots \cdot (-1)^{k_N(j)} \tag{2.26}$$

with $k_i(j) \in \{0, 1\}$ being the measurement outcome of qubit i in the jth run.

Measuring expectation values of observables is crucial for many quantum algorithms such as the VQE introduced in [Peruzzo et al., 2014], which is one of the most prominent quantum computing applications in the NISQ era. Its goal is to determine the ground state of a physical system, but various other optimisation problems can be mapped to this scenario. We take VQE as our main motivation for certain choices of quantum circuits and quantities of interest later in this thesis.

2.2. Quantum Noise

The established term NISQ era describing this time already points to one main characteristic of current quantum devices: They are prone to noise. Running a quantum circuit on hardware of this generation will yield incorrect results, making reliable quantum computations challenging. Although noise levels are decreasing [Tannu and Qureshi, 2019], current error rates do not allow for fault-tolerant quantum computing [Aharonov and Ben-Or, 2008; Knill, Laflamme, and Zurek, 1998; Kitaev, 2003; Shor, 1996] yet.

We can demonstrate the noisy behaviour of quantum computers by running the quantum circuit from Figure 2.3 on a NISQ device and comparing the results to the ideal outcome. As explained in Section 2.1.3, the ideal execution of this quantum circuit should always yield the measurement result '11'.

In contrast, Figure 2.9 shows the results from running the same quantum circuit $s = 1024$ times on IBM's `ibmq_manila` device. Since this only serves as an instructive demonstration, we do not discuss the details of this device here but refer to later chapters, where the same hardware is used to perform several quantum experiments. The results deviate from our ideal predictions, and the states $|00\rangle$, $|01\rangle$, and $|10\rangle$ were measured, although this should not be possible on a perfect quantum computer. The causes of such errors in quantum computers are manifold, as we explain in the following.

Creating stable qubits and entanglement between them is very challenging. They must be perfectly isolated from their environment to preserve their state, except for the exact moment of applying a quantum gate, which requires a short and controlled interaction. Qubits with a weak coupling to their environment are hard to control or measure, while easily accessible qubits are less stable. A balance between the two traits must be found,

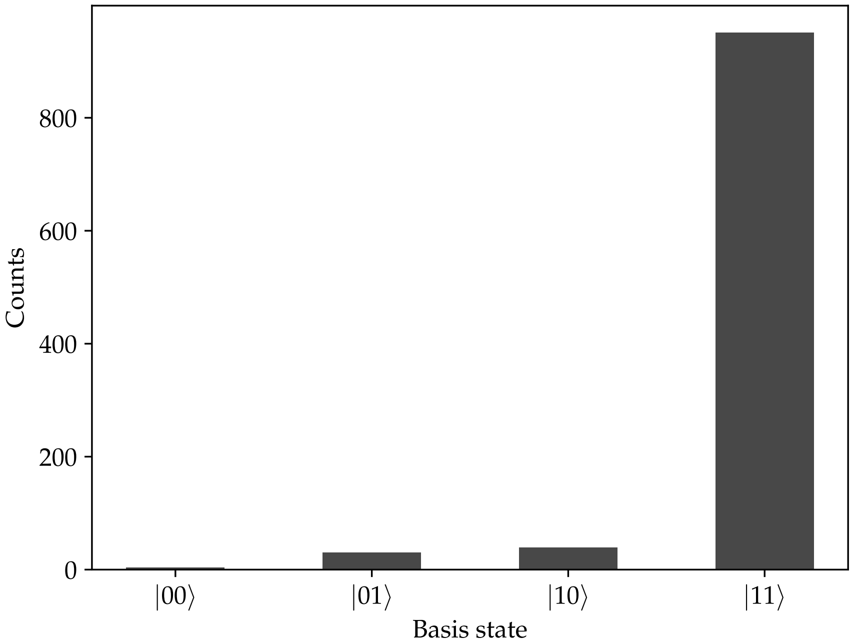


Figure 2.9.: Histogram with noisy results.

leading to imperfections that cause quantum noise or *decoherence*.

When qubits interact with their environment, unintentional quantum mechanical measurements take place, leading to a loss of information called depolarisation [Barnes et al., 2017]. The stability of qubits is usually quantified by two times T_1 and T_2, which correspond to two primary processes causing errors: thermal relaxation and dephasing. If a qubit is prepared in the excited state $|1\rangle$, it experiences a decay towards the ground state $|0\rangle$ over time, and the relaxation time T_1 measures this decay [Resch and Karpuzcu, 2021]. On the other hand, T_2 describes the transition of a quantum state towards a classical one.

While qubits are intended to interact with each other to create entanglement, unwanted qubit-qubit interactions are inevitable. They are generally called crosstalk and can lead to erroneous quantum computation. Crosstalk refers to all correlations between qubits not intentionally created by a quantum circuit. Additionally, quantum gates should be local, meaning that a gate acting on a single qubit should not influence others (except if they are entangled). Violations against this locality are also a form of crosstalk. Since crosstalk can be highly diverse, it is hard to detect and characterise. A comprehensive discussion on this subject can be found in [Sarovar et al., 2020].

Besides unwanted interactions of qubits with their environment or each other, imperfect implementations of circuit operations are a primary cause of quantum noise. The preparation of measurement of an incorrect qubit state leads to SPAM error, and quantum gates do not provide the desired control. For example, the rotation gates from 2.1.2 often over- or under-rotate the qubit, leading to co-called coherent errors. Finally, many physical realisations of qubits actually consists of more than the two computational basis states $|0\rangle$ and $|1\rangle$. If a qubit transitions to another state, it might become impossible to correctly control or measure this qubit for the rest of the computation. This phenomenon is called leakage.

Understanding all interactions and processes taking place in a quantum computer and understanding their sources is often impracticable. Instead, one tries to describe the effects on computations to incorporate quantum error mitigation. A specific formalism is typically used for representing quantum noise, and this section introduces noise models and all necessary preliminaries. Noise models describe and predict the noisy behaviour of quantum computers. They are based on an alternative description of quantum systems called density matrices. While the state vectors from Section 2.1.1 are insufficient to model noisy processes in quantum computing [Wilde, 2017], density matrices are a powerful tool to represent such processes. After discussing these density matrices in general, we introduce various types of quantum noise that typically occur on NISQ devices.

2.2.1. Density matrices

Section 2.1.1 provides a means of representing the state of a qubit as a two-dimensional vector $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$. The superposition of the two basis states makes a measurement in this basis probabilistic. However, this does not mean that the state vector $|\psi\rangle$ itself is probabilistic; on the contrary, qubits are in a well-defined overall state at each point of a quantum circuit. This is no longer the case if noisy processes occur. A qubit might be in state $|\psi_i\rangle$ with probability p_i (which is not the same as a superposition of these states). The collection $\{|\psi_i\rangle, p_i\}$ is called an *ensemble* of states. Therefore, another formalism is needed to describe quantum systems interacting with their environment. In the following, we define *density matrices* or *density operators* (we use these terms interchangeably) and explain how they overcome the abovementioned limitations. From now on, let $\mathcal{B}(\mathcal{H})$ be the space of bounded linear operators $B\colon \mathcal{H} \to \mathcal{H}$.

Following [Hall, 2013], a density matrix of a (finite-dimensional) quantum mechanical system with Hilbert space $\mathcal{H}$ is a linear operator $\rho \in \mathcal{B}(\mathcal{H})$ such that

1. ρ is Hermitian: $\rho = \rho^\dagger$,
2. ρ is positive semi-definite, i.e., all eigenvalues of ρ are non-negative, and
3. $\mathrm{Tr}(\rho) = 1$.

Density operators no longer represent the state of N qubits as a 2^N-dimensional vector but as a $2^N \times 2^N$ matrix. The usual state vector representation is a special case of this and can be translated to the density operator formalism in the following way. Given a set of qubits in state $|\psi\rangle$, one can construct their density matrix as

$$\rho = |\psi\rangle\langle\psi| \ . \tag{2.27}$$

Again, see Appendix A for more details on the Dirac notation. The action of a unitary gate U on the state ρ is given by

$$\rho' = U\rho U^{-1} \, . \tag{2.28}$$

Consider the case of an ensemble $\{|\psi_i\rangle \, , p_i\}$ of states again. The following result explains how density operators can mathematically describe such an ensemble. A proof can be found in [Nielsen and Chuang, 2010].

Proposition 2.2.1. *Every ensemble $\{|\psi_i\rangle \, , p_i\}$ gives rise to a density matrix*

$$\rho = \sum_i p_i \, |\psi_i\rangle\langle\psi_i| \ . \tag{2.29}$$

Conversely, for every density matrix ρ, one can find an ensemble $\{|\psi_i\rangle \, , p_i\}$ such that (2.29) holds. This choice is generally not unique.

Therefore, density operators directly correspond to ensembles of quantum states and are a natural generalisation. If a system is in state ρ_i with probability p_i, its density matrix is

$$\rho = \sum_i p_i \rho_i \, . \tag{2.30}$$

The Bloch sphere and pure vs mixed states

To better understand density matrices, we briefly discuss the case of a single qubit. The basis states $|0\rangle$ and $|1\rangle$ correspond to the 2×2 matrices

$$|0\rangle\langle 0| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad |1\rangle\langle 1| = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} , \tag{2.31}$$

respectively. Since the Pauli matrices $\{I, X, Y, Z\}$ form a basis of Hermitian 2×2 matrices (I denoting the identity matrix), every density matrix of a single qubit can be written as a linear combination of these Pauli matrices. However, not every linear combination results in a valid density matrix due to the conditions 1.-3. from above. Following [Nielsen and

Chuang, 2010], it turns out that this linear combination takes the form

$$\rho = \frac{I + r_x X + r_y Y + r_z Z}{2}\,, \tag{2.32}$$

where $\vec{r} = (r_x, r_y, r_z)^T$ is a three-dimensional real vector with $\|\vec{r}\| \leq 1$ that characterises the qubit and corresponds to the vector in the Bloch sphere from Figure 2.1. For example, we can directly confirm that the vector $r = (0, 0, 1)^T$ corresponding to the north pole of the Bloch sphere also results in the density matrix $|0\rangle\langle 0|$ from (2.31). Note that every point within the Bloch sphere is a valid density operator and not just the surface. This difference leads us to the notion of pure and mixed states. A state ρ is called *pure* if there is a vector $|\psi\rangle \in \mathcal{H}$ such that $\rho = |\psi\rangle\langle\psi|$. Otherwise, ρ is called a *mixed* state. If quantum computers worked perfectly, there would be no need to introduce density operators in this thesis. If a set of qubits is initialised as $|\psi\rangle = |0 \ldots 0\rangle$, the initial state is pure. It remains a pure state after applying quantum gates without noise. However, this changes in the presence of quantum noise, as explained in Section 2.2. The following criterion is useful to determine whether an arbitrary density operator is pure or mixed.

Proposition 2.2.2. *A state ρ is pure if and only if* $\mathrm{Tr}\left(\rho^2\right) = 1$.

In the special case of a single qubit, the pure states are exactly the ones located on the surface of the Bloch sphere. Equivalently, writing the density matrix as in (2.32), ρ is pure if and only if $\|\vec{r}\| = 1$. The (normalised) identity matrix $\rho = I/2$ is an example of a mixed single-qubit state. We can briefly confirm this by computing

$$\mathrm{Tr}\left(\rho^2\right) = \mathrm{Tr}\left(\frac{I}{4}\right) = \frac{1}{2} \tag{2.33}$$

and applying Proposition 2.2.2.

Expectation values

Section 2.1.5 discusses quantum mechanical expectation values for state vectors. These expectation values can also be obtained from density matrices. If $O \colon \mathcal{H} \to \mathcal{H}$ is an observable, then its expectation value in state $\rho = \sum_i p_i \,|\psi_i\rangle$ is

$$\langle O \rangle = \sum_i p_i \langle \psi_i | O | \psi_i \rangle = \sum_i p_i \,\mathrm{Tr}(|\psi_i\rangle\langle\psi_i|\, O) = \mathrm{Tr}(O\rho)\,. \tag{2.34}$$

Composite systems

We conclude the discussion on density matrices by considering composite systems. As explained in Section 2.1.1, the Hilbert space of N qubits with spaces $\mathcal{H}_1, \ldots, \mathcal{H}_N$ is the tensor product $\mathcal{H} = \mathcal{H}_1 \otimes \ldots \otimes \mathcal{H}_N$. In the following, we only consider two qubits, but everything can be generalised to an arbitrary number. If the qubits are prepared in initial states $\rho^{(1)}$ and $\rho^{(2)}$, the composite system has the density operator $\rho^{(1)} \otimes \rho^{(2)}$. However, not every density operator ρ on $\mathcal{H}_1 \otimes \mathcal{H}_2$ can be written as a sum

$$\rho = \sum_i p_i \rho_i^{(1)} \otimes \rho_i^{(2)} , \tag{2.35}$$

where all $\rho_i^{(k)}$ are density operators of the kth qubit and $\sum_i p_i = 1$. In fact, a state as in (2.35) is called *separable* and the question whether it is separable or not is proven to be an NP-hard problem [Gharibian, 2010].

2.2.2. Noise channels

In the ideal quantum computing scenario from Section 2.1, a vector represents the state of a set of qubits, and unitary matrices describe quantum gates that change that state. After laying the mathematical foundation to replace state vectors with density matrices for the presence of noise, we also introduce maps that change these density matrices. This allows us to describe and understand the processes we encounter on NISQ hardware, including models for various types of errors that typically occur. We first provide the mathematical definition of quantum channels and explain afterwards why this definition makes sense for representing quantum noise.

Consider two quantum systems with Hilbert spaces $\mathcal{H}$ and $\mathcal{H}'$. In quantum computing, they often coincide and correspond to the same set of qubits, i.e., $\mathcal{H} = \mathcal{H}' = \mathbb{C}^{2N}$. Recall that a density matrix on $\mathcal{H}$ is a positive semi-definite linear operator $\rho\colon \mathcal{H} \to \mathcal{H}$ with trace equal to one. A quantum channel between the two systems is a linear operator $\mathcal{E}$ mapping density operators of $\mathcal{H}$ to density operators on $\mathcal{H}'$. It satisfies the following conditions:

1. $\mathcal{E}$ is completely positive.

2. $\mathcal{E}$ preserves the trace, i.e., $\mathrm{Tr}\left(\mathcal{E}(\rho)\right) = 1$ for a density operator ρ.

These two conditions above ensure that the result of applying $\mathcal{E}$ to a density matrix is again a valid density matrix. In the following, we present an important theorem that classifies all noise channels before considering some instructive examples.

Kraus' theorem

While the definition of noise channels as completely positive, trace-preserving (CPTP) maps is intuitive and relates to the properties of density matrices, it might not always help to understand their nature. The authors of [Choi, 1975; Kraus et al., 1983] present an alternative description of CPTP maps that renders very useful in the context of noisy quantum computers, particularly for simulating noise models.

Let $\mathcal{H}$ and $\mathcal{H}'$ be two Hilbert spaces (again, keep the quantum computing scenario $\mathcal{H} = \mathcal{H}'$ in mind) and consider a linear map $\mathcal{E}$ between the corresponding spaces of bounded operators. Then Kraus' theorem states the following:

Theorem 2.2.3 (Kraus' theorem [Choi, 1975; Kraus et al., 1983]). *$\mathcal{E}$ is a noise channel if and only if there are linear maps $E_i \colon \mathcal{H} \to \mathcal{H}'$ such that*

$$\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger \quad \textit{with} \quad \sum_i E_i^\dagger E_i = I\,. \tag{2.36}$$

We call the form above the operator-sum representation.

The condition on the left hand side of (2.36) ensures that $\mathcal{E}$ is completely positive and the sum on the right hand side corresponds to the trace-preserving property.

Example: quantum gates

In Section 2.2.1, we already discussed how quantum gates change the state of qubits in the density matrix formalism, see (2.28). This map can also be interpreted as a noise channel $\mathcal{E}(\rho) = U\rho U^{-1}$. We briefly show that is it trace-preserving by using the cyclicity of the trace and computing

$$\mathrm{Tr}\left(U\rho U^{-1}\right) = \mathrm{Tr}\left(U^{-1}U\rho\right) = \mathrm{Tr}(\rho)\,. \tag{2.37}$$

Alternatively, the unitary channel is obviously written as an operator-sum representation.

Example: depolarising channel

The depolarisig channel describes a maximal loss of information about the state of a quantum system. We define its action on N qubits as

$$\mathcal{D}_N(\rho) = (1-\lambda)\rho + \frac{\lambda}{D}\cdot I\,, \tag{2.38}$$

where λ is the probability of depolarisation and $D = 2^N$ is the dimension of the system. Hence, the density matrix ρ is replaced with the normalised identity matrix with probability λ. Using the Pauli matrices from (2.11), the depolarising error on a single qubit has the following operator-sum representation:

$$\mathcal{D}_1(\rho) = \left(1 - \frac{3\lambda}{4}\right)\rho + \frac{\lambda}{4}\left(X\rho X + Y\rho Y + Z\rho Z\right)\,. \tag{2.39}$$

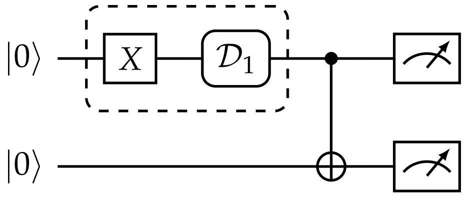


Figure 2.10.: Quantum circuit with noisy X gate subject to single-qubit depolarising error.

Therefore, the depolarising error on a single qubit can equivalently be described as acting with each of the Pauli matrices with probably $\lambda/4$. Note that some authors define the parameter λ differently, resulting in rescaled factors in the above equations.

2.2.3. Noise models

Noise channels provide a way to model errors that can occur during quantum computations. However, we do not only want to describe individual errors but the complete noisy behaviour of a quantum computer. Such a complete description is called a *noise model*. A noise model consists of a collection of noise channels and precise instructions on where they occur during the execution of a quantum circuit. For now, it can be interpreted as a map M that takes as input a quantum circuit c and maps it to a noisy final state ρ at the end of the circuit, predicting the erroneous result of the quantum circuit. However, we will refine the definition of a noise model in Chapter 5 of this thesis.

The building blocks of a quantum circuit are state preparation, quantum gates, and measurements. Technically, the idle operation is also part of these building blocks. A noise model assigns to each operation in a quantum circuit a noise channel applied directly before or after the affected operation. If multiple noise channels affect the same one, the noise model specifies the order in which they are applied.

For example, consider that only single-qubit gates are erroneous and affected by depolarising error. This can be modelled by applying a single-qubit depolarising channel $\mathcal{D}_1$ after each of these gates. The quantum circuit from Figure 2.3 has one single-qubit gate: the X gate. With this basic example of a noise model, every operation in that quantum circuit is assumed to be executed perfectly except for the X gate. Figure 2.10 shows the resulting quantum circuit after applying the noise model. A rounded box with a symbol for the noise channel $\mathcal{D}_1$ is added and the dashed lines indicate to which gate it corresponds. We discuss this graphical representation of noise models further in Chapter 5.

Most noise channels depend on parameters such as probabilities that specific errors occur. For example, the depolarising channel depends on the probability that the density operator is replaced by the identity matrix. Therefore, any noise model containing these channels also depends on a set of parameters, which we denote as θ.

Noisy simulations

Section 2.1.4 discusses the simulation of quantum circuits under ideal circumstances. One can equip this simulation with a noise model M to predict the noisy outcomes of quantum circuits. Again, quantum circuits can be simulated shot-wise or using exact computations based on density matrices. The latter is often feasible for reducing the variance due to finite shot numbers. If the goal is to predict noisy expectation values, (2.34) can be used to compute them directly from the density matrix.

2.3. Quantum Error Mitigation

Understanding and predicting the noisy behaviour of a quantum computer alone does not improve computational results. Methods to counteract errors are needed to enable reliable computations. Quantum error mitigation intends to approximate the ideal results of a quantum circuit from the erroneous ones using the understanding of quantum noise given by noise models. Various error mitigation methods exist that aim to tackle different types of error. Prominent examples are probabilistic error cancellation, zero-noise extrapolation, and measurement error mitigation. Several approaches have been proposed for the latter. Using the readout error model from [Funcke, Hartung, Jansen, Kühn, Stornati, and Wang, 2022], one can predict its impact on quantum mechanical expectation values, allowing for an estimate of the ideal result.

While quantum error mitigation can significantly improve the results of quantum computations, this improvement comes at a price. A large number of samples, i.e., the number of quantum circuits to be executed or the number of shots, is needed to counteract an increasing variance of results. [Takagi et al., 2022] provides a lower bound on these samples for a broad class of mitigation methods. In particular, the necessary number of samples grows exponentially with the number of gate layers in a quantum circuit when mitigating gate errors.In [Alexandrou et al., 2022], we investigated the variance increase for the measurement error mitigation from [Funcke, Hartung, Jansen, Kühn, Stornati, and Wang, 2022] using different quantum hardware.

Due to the sampling cost of quantum error mitigation, combining many methods at once is not feasible. Instead, a tradeoff between the improvement of results and the sampling cost can be made using targeted methods to mitigate the most dominant types of noise. Accurate noise models are crucial to allow for such prioritisation of quantum error mitigation methods.

2.4. Benchmarking

This section discusses benchmarking fundamentals and introduces essential aspects to improve the quality of a benchmark. We define a benchmark as a test for *evaluating and comparing* systems, tools, or techniques with respect to specific *characteristics*. While this is initially inspired by work on computer benchmarks [Kistowski et al., 2015; Kounev, 2021; Sim et al., 2003], the definition can easily be adapted to other settings. It can be used to assess quantum hardware, to test the effectiveness of error mitigation methods, or to evaluate and compare the accuracy of different noise models, which is the main focus of this thesis. A benchmark usually consists of a workload that must be processed. For example, the workload of a CPU performance benchmark could consist of computational tasks to be solved by the tested CPUs.

Quality attributes

To ensure that the results of a benchmark provide helpful information, several desirable properties have been developed that benchmarks should have. Kistowski et al., 2015 summarise these properties in five categories that we use as quality attributes and that we discuss in the following.

- **Relevance:** A benchmark is relevant if its results generalise well to real-world applications of interest. Achieving high relevance requires the choice of a workload that can capture the tested characteristics. Note that relevance always depends on the intended application context.

- **Reproducibility:** The results of a reproducible benchmark remain consistent over multiple runs, given the same configuration. Reproducibility refers to multiple runs in the same environment and runs by different users in possibly different environments.

- **Fairness:** A benchmark is fair if it does not impose artificial constraints on the system under test. It should be unbiased, and no benefits or punishments for different competitors should stem solely from the benchmark design.

- **Verifiability:** The verifiability of a benchmark ensures it is performed correctly and all instructions are respected. Self-validation is a possible measure for improving verifiability.

- **Usability:** A benchmark is usable if it is easy to run by a user. The necessary hardware and software configuration for the benchmark should be straightforward to obtain.

So far, these attributes have been formulated in general terms and can be applied to any field. Later we concretise them for benchmarking noise models.

CHAPTER

THREE

RELATED WORK

This chapter discusses relevant publications related to the content of the thesis. It is structured based on the three research questions introduced in Chapter 1 and the corresponding research fields. Section 3.1 examines work related to the visualisation of quantum noise, followed by a discussion of literature concerning benchmarks in Section 3.2. The final Section 3.3 addresses the characterisation of noise and the construction of noise models.

3.1. Visualising Quantum Noise

This thesis presents an extension of quantum circuits by noise channels. The quantum circuit model was initially introduced by Deutsch, 1989 and is the prevalent way to describe quantum computation. Yao, 1993 proved its equivalence to an alternative model, the quantum Turing machine. A quantum circuit consists of initial qubit state preparation, quantum gates, and measurement. Qubits and gates are represented by horizontal wires and boxes, respectively (some gates are exceptions and receive special symbols). Moreover, measurements are depicted using a meter symbol, as shown in Figure 3.1. All quantum gates are unitary matrices applied to the state vector of the qubits. But quantum channels are generally not unitary, and adding them to the quantum circuit model requires a new interpretation of quantum circuits. This interpretation must be consistent with the mathematical formalism of noise channels. Choi, 1975 and Kraus et al., 1983 proved an essential classification of noise channels. In particular, we use it to model channels restricted to subsets of qubits. In summary, our work concerning noise model visualisation primarily consists of two parts: the graphical representation and the new meaning of quantum circuits.

While several publications show figures of quantum circuits containing symbols for errors, almost no work explicitly addresses visualising noise models by formally intro-

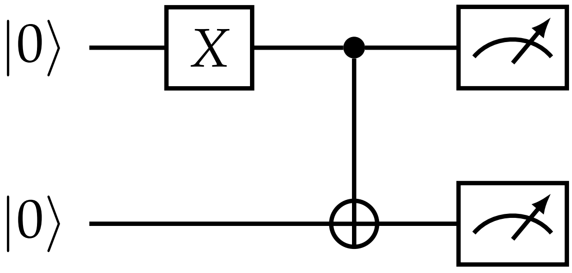


Figure 3.1.: Basic two qubit quantum circuit.

ducing errors to the circuit model. Aharonov, Kitaev, et al., 1998 generalise the quantum circuit model to density matrices, where quantum gates can be arbitrary quantum channels. They define quantum circuits with mixed states as directed acyclic graphs with quantum channels at the nodes and use these graphs as visualisation. While our approach is similar in extending quantum circuits to density operators and quantum channels, there are significant differences. For example, our visualisations are entirely different and preserve existing quantum circuits. We discuss the similarities and differences between the approaches further in the corresponding Chapter 5.

Ahsan et al., 2022 present a method to counteract a specific type of crosstalk in quantum computers by introducing additional gates to cancel out the generated noise. These gates are highlighted in the quantum circuits to point to the locations where crosstalk arises. Chancellor et al., 2023 introduce a graphical calculus for designing and analysing error correction codes. Although the calculus is not related to the quantum circuit model, several circuits with boxes representing arbitrary quantum noise are shown in the work. They do not correspond to any specific noise model and are only used to demonstrate a procedure to detect errors called coherent parity check. Trout et al., 2018 simulate a quantum error correction code using a noise model for ion trap quantum computers. They graphically represent the effect of their noise model on quantum gates with a figure similar to the ones we produce in this thesis. However, this serves as a basic illustration and is not an approach to adding errors to the circuit model. Finally, Emerson et al., 2007 introduce a method for characterising noisy quantum processors. In their work, they depict a noisy quantum circuit. The included noise channels always affect all qubits, and their appearance in the circuit model is not being further discussed. All the publications considered above use some graphical representations of quantum noise. Still, in contrast to this thesis, they do not consider noise channels a formal part of the circuit model or an extension thereof.

In this thesis, we are primarily concerned with noise models in the sense of quantum channels applied to density operators. There are other formalisms to describe the noisy processes of quantum computers, such as master equations [Lindblad, 1976; Breuer and Petruccione, 2007] or stochastical differential equations [Plenio and Knight, 1998; Jacobs and Knight, 1998]. Bassi and Deckert, 2008 introduce another approach called noise gates. Noise gates represent quantum noise without requiring density operators and can be visualised in quantum circuits as gates applied before or after the original quantum gates.

Visualising noise models does not necessarily imply using the quantum circuit model. While the circuit model gives an insight into where errors occur and which operations are affected, other views on quantum noise and noise models can also provide valuable information. Berg et al., 2023 visualise noise models in terms of a qubit layout and other diagrams containing information on the error parameters and gate fidelities. Ruan et al., 2023 present a similar but more extensive visualisation approach consisting of multiple views. For example, the authors compute quantities such as best or worst gate performance and other metrics that are plotted.

Besides gate-based quantum computing as described in this thesis, there are other methods to realise quantum computation, such as topological quantum computers, which were introduced by Kitaev, 2003. For this type of quantum device, Knapp et al., 2018 present noise models and error correction methods. Since the computation fundamentally differs from gate-based quantum computing, the visualisations of quantum noise also differ from the ones we propose.

3.2. Benchmarking Noise Models

One of the main contributions of this thesis is a volumetric benchmarking framework for evaluating and comparing quantum computing noise models regarding their accuracy. It involves computing the deviation between noise model predictions and hardware results for a collection of test quantum circuits of varying sizes using an objective function. While only a few publications exist on benchmarking the capability of error models to describe hardware behaviour, many approaches assess the fidelity of quantum devices. The difference is that these approaches compare hardware results to the ideal ones instead of model predictions. In both cases, the outcomes of quantum circuits must be compared with respect to specific loss functions. Since much can be learned from the quantum hardware benchmarking literature concerning these properties, we include the most relevant work in our discussion here.

This section is further divided into the following subsections. In 3.2.1, we discuss work directly related to volumetric benchmarks. The relevant literature on existing approaches to evaluating noise models is discussed in 3.2.2. Afterwards, we discuss publications concerning the quantum circuits and objective functions to use for benchmarks in 3.2.3. Finally, 3.2.4 briefly summarises related work on the benchmark quality attributes used for our evaluation in Chapter 6. Note that publications might be mentioned in multiple sections if they contribute to more than one category.

3.2.1. Volumetric Benchmarks

Volumetric benchmarks have been initially introduced by Blume-Kohout and Young, 2020 to test quantum computers regarding their reliability for specific applications. They compare hardware results to ideal outcomes for quantum circuits of varying sizes or *volumes*. These volumes are characterised by two parameters called depth d and width w, which correspond to the numbers of gates (or layers thereof) and qubits, respectively. The authors suggest several success criteria for the comparison, which must be computed from the outcome probability distributions. Besides direct distance measures of the distributions, such as Hellinger distance, one can use quantum circuits with well-defined correct output states and calculate probabilities of obtaining this state.

Since their proposal, the first volumetric benchmarks have been performed on NISQ hardware. Proctor, Rudinger, Young, Nielsen, et al., 2022 report volumetric benchmark results for twelve quantum processors. Since the ideal outcomes of the used quantum circuits must be known, the capability to compute them efficiently is essential. The authors address this problem with mirror circuits. They are constructed by mirroring base quantum circuits and applying their inverses, preserving important properties while being efficiently predictable. Using these mirror circuits, volumetric benchmarks are performed for several quantum devices and simulations are carried out based on reported error rates by the providers. The publication finds that they are not sufficient to predict the success probabilities of the test quantum circuits, emphasising that crosstalk must be considered.

Lubinski et al., 2023 perform modified volumetric benchmarks using typical quantum circuits for several quantum algorithms, such as Bernstein-Varizani, Deutsch-Josza, and Grover's Search [Bernstein and Vazirani, 1997; Deutsch and Jozsa, 1997; Grover, 1996]. Besides the deviation between hardware and ideal results, they also report the execution times of the circuits, allowing for better estimations of the feasibility of running a quantum algorithm on a machine. Cirstoiu et al., 2023 propose a volumetric approach to benchmark quantum error mitigation. They compare the ideal outcomes of quantum circuits to the hardware results after applying different mitigation methods. The benchmarks are performed on current quantum devices for zero-noise extrapolation and Clifford data regression [Czarnik et al., 2021], and the deviation between two circuit outcomes is computed using quantum mechanical expectation values.

3.2.2. Other Approaches

In the following, we summarise the literature evaluating quantum computing noise models. Such evaluations are rarely the main subject of a publication but are often presented with new noise models or methods to generate them. Dahlhauser and Humble, 2021 introduce a procedure to develop error models by decomposing training circuits into subcircuits and repeatedly updating the model according to its predictions on these subcircuits. They evaluate the resulting noise model by comparing outcomes of noisy simulations to hardware data with respect to the total variation distance. The quantum circuits are typical circuits for the Bernstein-Varizani algorithm and for preparing the so-called Greenberger-Horne-Zeilinger state. In a more recent publication [Dahlhauser and Humble, 2022], the same authors expand this evaluation to noise models resulting from other characterisation methods. Their approach from above is compared to gate set tomography, randomised compiling, and noise reconstruction using similar quantum circuits. The results show that noise model performance depends on the application and the corresponding quantum circuits. For a discussion of the tomography methods above, we refer to Section 3.3.2.

Georgopoulos et al., 2021a present an alternative approach for producing a noise model. The model includes state preparation and measurement (SPAM) error with symmetric probabilities, depolarising error, and thermal relaxation/dephasing. Its parameters are obtained through hardware calibration or a genetic learning algorithm. Both resulting noise models are evaluated by computing the Hellinger distance [Hellinger, 1909] between outcome distributions of noisy simulations and hardware experiments. Quantum random walk circuits are used for the evaluation (see [Aharonov, Davidovich, et al., 1993]). Moreover, the authors test similar noise models, including the Qiskit device model. The results show that the parameters optimised with the genetic algorithm yield better predictions than the calibrated ones provided by IBM. In another publication, Georgopoulos et al., 2021b also perform extended benchmarks of their noise model, using quantum circuits from several quantum computing applications and three different IBM devices for evaluation. The average Hellinger distance is computed between the outcome distributions of hardware results and ideal/noisy simulation.

Ketterer and Wellens, 2023 discuss crosstalk errors and construct a crosstalk-aware noise model based on the Qiskit device model. They use randomised benchmarking for characterising the noise and compare their extended noise model to the vanilla Qiskit one. The objective function is the Hellinger fidelity, which is closely related to the Hellinger distance from above, and the quantum circuits consist of Hadamard gate ladders. The authors find that noise models should not neglect crosstalk errors because they substantially impact quantum computation.

Finally, Blume-Kohout, Rudinger, et al., 2020 present a technique to quantify unmodelled errors of a given noise model by introducing a wildcard model that accounts for these errors. The wildcard model assigns error rates to all gate and SPAM operations to compensate for imperfections of the original noise model. The authors show the success of this procedure for a model obtained by gate set tomography, significantly improving the model accuracy. However, it does not give insight into the underlying physical processes.

3.2.3. Quantum Circuits and Objective Functions

Every experimental evaluation of a noise model involves computing the deviation between two quantum computing results (for example, model prediction and hardware data) for a collection of quantum circuits using an objective function. Gilchrist et al., 2005 comprehensively discuss the feasible properties of these objective functions concerning computability, relevance, and other motivations. Moreover, they present possible choices and their respective advantages and disadvantages. Numerous measures between quantum processes and techniques to practically calculate them from quantum experiments are explained.

3.2.4. Quality Attributes

The literature on systems benchmarking discusses desirable properties of a benchmark. Although they refer to testing classical computing systems, many of these properties also apply to quantum computing, particularly for benchmarking noise models. Kounev, 2021 and Kistowski et al., 2015 categorise them into five categories we use as quality attributes for designing volumetric benchmarks. Sim et al., 2003 discuss similar criteria with a slightly different focus.

3.3. Constructing Noise Models

Characterising noise in quantum computers is the subject of many publications. Various approaches exist for constructing noise models for several types of quantum computers with different architectures. As explained earlier, this thesis focuses on gate-based quantum computing. Moreover, we work with noise models in terms of noise channels represented by CPTP maps on density operators. In the following, we summarise related literature on characterising errors within the abovementioned restrictions, with some exceptions. We further divide our summary into several subsections. Firstly, we discuss publications directly related to our approach to constructing noise models in 3.3.1. This includes existing methods to optimise noise model parameters. Secondly, we give an overview of quantum tomography techniques in 3.3.2. Finally, 3.3.3 presents alternative approaches to constructing noise models and other relevant publications.

3.3.1. Optimising Model Parameters

Our noise model constructed in Chapter 7 contains five types of noise affecting all quantum circuit operations. State preparation is considered erroneous, and measurements are susceptible to asymmetric readout error. Furthermore, gates are prone to thermal relaxation and dephasing, crosstalk, and depolarising error. The noise model depends on various parameters, such as the bit-flip probabilities of the readout error. These parameters are optimised using noisy simulations and comparing them to quantum hardware data. After each iteration, they are updated according to the simultaneous perturbation stochastic approximation (SPSA) algorithm [Spall, 1992] or a least-squares fit.

As mentioned in 3.2.2, Georgopoulos et al., 2021a define a similar noise model that only uses symmetric readout error and does not include crosstalk. The model parameters are partially optimised by a genetic algorithm (see [Mitchell, 1996]), and the remaining parameters are obtained from device calibrations. Berg et al., 2023 define a noise model consisting of Pauli noise channels. It depends on the fidelities of the Pauli matrices that are learned by running quantum circuits of varying depths, measuring the overall fidelities, and applying an exponential fit to the experimental data. Dahlhauser and Humble, 2021 divide quantum circuits into subcircuits and use noise gates from Bassi and Deckert, 2008 to model the noise in each subcircuit. The noise gates depend on error rates that are optimised with Powell's hybrid method [Powell, 1970]. Majumder et al., 2023 develop a noise model for quantum computers based on ion traps. It includes over-rotations, phase error, and detuning error affecting quantum gates. Again, the model parameters are fitted to experimental data, where the authors use a least-square fit.

3.3.2. Quantum Tomography

In recent years, various quantum tomography techniques have been developed that aim to characterise quantum noise by performing specific hardware experiments. In the following, we discuss the relevant approaches for this thesis and explain what they measure. A comprehensive overview of most methods can be found in [Greenbaum, 2015].

One of the earliest tomography methods is quantum state tomography, introduced by Vogel and Risken, 1989. Its goal is to characterise an unknown quantum state by measuring different components of the density operator. It can be used to describe quantum noise by comparing the actual quantum state to the desired one. While this allows for a first estimate of the fidelity of a quantum device, more detailed characterisations of quantum noise are needed [Turchette et al., 1995]. Chuang and Nielsen, 1997 and Poyatos et al., 1997 suggest determining quantum processes, i.e., the dynamics changing the state of qubits, using quantum process tomography. Quantum process tomography can describe noisy quantum gates by preparing different states ρ_i, applying the tested gate, and performing specific measurements. However, a noise channel on N qubits is generally represented by an $4^N \times 4^N$ matrix, so the number of necessary experiments grows exponentially.

To overcome the unfeasible scaling of quantum process tomography, Emerson et al., 2007 and Knill, Leibfried, et al., 2008 propose randomised benchmarking to obtain information on the fidelities of quantum gates. Randomised benchmarking circuits apply gate sequences of varying depths followed by their inverses. Since the ideal outcome of these circuits is the initial state, the deviation of the experimental results from the expected ones can be evaluated efficiently. The gate fidelities are then estimated using an exponential fit. Proctor, Rudinger, Young, Sarovar, et al., 2017 provide a comprehensive discussion on the accuracy of this estimate. Cycle benchmarking [Erhard et al., 2019] and character benchmarking [Helsen et al., 2019] are two prominent derivatives of randomised benchmarking that address its dependence on specific quantum gates called Clifford gates.

While quantum process tomography intends to describe the noise of a single gate, Gate set tomography aims to characterise all circuit operations, including state preparation and measurement. This method was initially proposed by Merkel et al., 2013, but other authors independently derived similar protocols at the same time [Kim, Ward, et al., 2015; Blume-Kohout, Gamble, et al., 2013]. Although it offers a thorough model of the noise in a quantum device, its scaling makes it unfeasible for large qubit numbers. We do not discuss gate set tomography in more detail here but refer to [Nielsen, Gamble, et al., 2021] for further information on the algorithm. Harper et al., 2020 introduce a more efficient characterisation protocol. They assume that the noise has a particular form and that noise channels are given by Pauli operators. This assumption is based on a publication by Wallman and Emerson, 2016 that presents a technique called randomised compiling, which transforms errors into these Pauli channels. The restriction drastically reduces the number of hardware measurements needed to obtain the error rates of the individual channels.

All the abovementioned methods assume that the noise in a quantum computer is time-independent. This is, however, often not the case, and error rates change over time. Proctor, Revelle, et al., 2020 propose a technique for detecting and measuring time-dependent noise in quantum hardware. It uses a time-dependent error model fitted to data obtained from time-resolved benchmarks.

3.3.3. Other Characterisation Approaches

As discussed in Section 2.1.4, various physical realisations of quantum computers have been suggested. Each variant might bring specific types of noise due to the unique architectures. In a recent publication, Gustiani et al., 2023 developed a virtual quantum device platform to emulate the different quantum hardware types. They present characteristic noise models based on typical physical processes for each device, allowing for architecture-specific noise simulations. Several authors also discuss quantum noise related to the physical realisations. Trout et al., 2018 consider various error types for ion-trap quantum computers, such as mode heating and photon scattering. Ghosh et al., 2013 concern the CZ gate for superconducting qubits, an alternative to the CX gate used in this thesis. They present a noise model based on the pulses by which the gate is physically controlled. Li, Ahmed, et al., 2022 also focus on these pulses and provide a pulse-based simulation tool for superconducting qubits and spin chains. Finally, Cattaneo et al., 2021 present a collision model that simulates quantum noise based on fundamental interactions (collisions) between qubits with each other and their environment.

CHAPTER

FOUR

RESEARCH OVERVIEW

This chapter provides an overview of the research presented in this thesis. Section 4.1 defines the goal and the research questions, while Section 4.2 summarises the contribution.

4.1. Goal and Research Questions

This section defines the goal of the thesis and presents the research questions investigated in the later chapters. Some of the research questions are divided into subquestions, also explained in this section.

Our primary research goal is the following:

> **Construct an understandable noise model that is more realistic than existing noise models from the literature.**

Understanding errors and their impact on quantum computations is crucial to counteract them with methods such as quantum error mitigation. Therefore, noise models must be constructed to describe and predict erroneous hardware behaviour. The statement of our goal contains several essential elements that we highlight in the following.

Firstly, the noise model should be understandable, i.e., formulated in a way that allows us to understand which quantum circuit operations are affected by errors. Since the quantum circuit model cannot represent non-unitary processes such as noise, it is natural to find an appropriate extension of the model to address this. Moreover, the noise channels included in the noise model should correspond to physical processes that can give us further insight into the causes of noise.

Secondly, a benchmarking framework to evaluate the accuracy of noise models should be developed to compare different models. Currently, no systematic approach for such an evaluation exists, making it nearly impossible to estimate the usefulness of a noise model. The framework should also consider the quantum computing application context because the accuracy of noise models might vary for different use cases.

Finally, the noise model should outperform existing competitors regarding the benchmarks from above. It can then be used for realistic quantum simulations when quantum computing resources are limited or to use targeted quantum error mitigation aimed at the most impactful types of noise.

From this primary goal, we derive the following three research questions with corresponding subquestions.

RQ1: How can noise models and their impact on quantum circuits be visualised?

Besides quantum error mitigation, optimising quantum circuit design can reduce the noise in quantum computations. For example, shallow circuits with few gates are less prone to gate error than deeper circuits. Visualising quantum noise and understanding which operations are affected the most can be beneficial for designing more robust quantum circuits. However, the quantum circuit model in its original form cannot represent noise channels but only unitary processes. Therefore, our research addresses the following subquestions:

RQ1.1: How can the quantum circuit model be extended to include noise channels?

Since the quantum circuit model is a widely used graphical representation of quantum computing, it is feasible to use this existing model and adapt it to a noisy setting. This adaption needs to include visualisations for noise channels that clearly indicate which circuit operation they influence. However, noise channels are entirely different mathematical objects than quantum gates and the interpretation of a quantum circuit as matrices acting on vectors representing qubits cannot persist. This research question addresses this challenge and explores the possibility of rigorously extending the circuit model by errors.

RQ1.2: To what degree can the extended circuit model generate noisy simulations?

Quantum circuits can directly be processed and executed by a quantum computer or a simulator. Since simulations with noise models are crucial for many applications, it would be desirable if such noisy simulations could be triggered directly from quantum circuits that include noise channels. Therefore, this question concerns the interplay between the circuit model and simulation.

RQ2: How can noise models be evaluated and compared regarding their accuracy?

As mentioned earlier, accurate noise models are essential for realistic quantum computing simulations and prioritising quantum error mitigation methods. Knowing whether a noise model can describe the noisy behaviour of quantum devices well requires a systematic way of benchmarking. This research question investigates the application of benchmarking frameworks for testing and comparing noise models. In particular, it addresses the following subquestions.

RQ2.1: Which benchmarking framework can provide valuable information on the accuracy of noise models for specific applications?

This question explores a procedure to measure noise model accuracy reliably with results generalisable to real-world applications.The quantum circuits used for a computation strongly depend on the application. They can differ in size and the quantum gates that are included. Since the noise in a quantum device could affect this diversity of quantum circuits to varying extents, noise models might be differently accurate as well. Therefore, this question addresses also the dependence of benchmarks on quantum computing applications.

RQ2.2: How can the benchmark quality be ensured regarding the attributes from Section 2.4?

Section 2.4 discusses desirable properties of benchmarks called quality attributes. They were initially specified for benchmarks of classical computing systems, and some thought is required to adapt them to the quantum world. This research question investigates the challenges for noise model benchmarks to fulfil the attributes and possibilities to overcome these challenges.

RQ3: How can realistic noise models be constructed?

This question addresses our primary research goal from above. It can be further divided into the following subquestions.

RQ3.1 What types of noise should an accurate noise model include?

Section 2.2.2 introduces the depolarising channel as one example of gate error. Similarly, various other types of noise exist in current quantum devices, and this question investigates models for different errors and a way to compose them into an overall noise model.

RQ3.2 How can noise model parameters be optimised?

Noise models typically depend on a set of parameters that must be determined to describe quantum noise accurately. For example, the depolarising channel depends on the depolarisation probability. Calibration experiments are one possibility for obtaining such parameters, but they are performed on noisy hardware, making them prone to error. Hence, they must be planned wisely to give reasonable estimates of the parameters. This research question explores an alternative approach to learning noise model parameters.

4.2. Summary of the Approach

Our approach to answering the above research questions consists of three parts. Firstly, we present an extension of the quantum circuit model called noisy quantum circuits (NQCs) in Chapter 5. An example of these extended circuits has already been shown in Figure 2.10, where noise channels are represented by boxes with rounded corners. Dashed lines indicate which operations are affected by noise. Furthermore, one can easily derive noisy simulations because every gate or noise channel can be written as a CPTP map and applied to the current density matrix.

Secondly, we propose volumetric benchmarks to evaluate and compare noise models in Chapter 6. Volumetric benchmarks have initially been introduced by Blume-Kohout and Young, 2020 to test quantum hardware. They consist of running test quantum circuits of different sizes or volumes on a quantum computer and comparing the results to the ideal ones. We transfer this idea to noise models, where the predictions of noise models are compared to hardware results. The choice of test circuits and different options for this comparison are discussed regarding their impact on the quality attributes.

Finally, we construct a noise model in Chapter 7 that considers all operations in a quantum circuit to be erroneous. It models state preparation error as a quantum bit-flip acting on the initial state and measurement error as a classical bit-flip after the measurement. The latter can have asymmetric probabilities. Besides SPAM error, the noise model includes gate error in the form of depolarisation, thermal relaxation and dephasing, and crosstalk. The parameters of this model are optimised by repeatedly comparing noise model predictions to hardware data for specific training circuits. Different loss functions and optimisation algorithms are discussed. Moreover, we analyse the parameters of the trained noise models regarding their drift over time and investigate the impact of different errors in the model on quantum computations.

CHAPTER

FIVE

VISUALISING NOISE MODELS

Before we construct and benchmark quantum computing noise models in the later chapters of this thesis, we establish a way to represent these noise models graphically. While the most accurate description of quantum noise is given by the mathematical formalism of noise channels, i.e., certain maps of density operators, graphical models can be more intuitive for understanding the underlying processes. This chapter presents our approach to extending the quantum circuit model with arbitrary quantum channels and is structured as follows. It is partially based on our work published in [Weber, Riebisch, et al., 2021].

Section 5.1 gives an introduction to the visualisation of noise and the quantum circuit model. It recalls our related research questions and discusses them in more detail. In Section 5.2, we present our approach and explain how quantum channels can be modelled in quantum circuits. Section 5.3 concerns the practical advantages of noisy quantum circuits for simulations. After evaluating our approach in Section 5.4, we finally give brief summary of noisy quantum circuits in Section 5.5.

5.1. Introduction and Research Questions

Quantum computing is a rapidly evolving research field with a constantly increasing number of participants. These participants come from various backgrounds, such as physics, chemistry, and computer science. They can have widely diverging knowledge on quantum noise, which is a complex subject, and fully understanding noise channels requires a sound mathematical foundation. However, noise cannot be neglected for practical quantum computations in the NISQ era. Addressing quantum errors with error mitigation methods is inevitable, and every researcher must be aware of the noisy nature of current machines. Visualising quantum noise could benefit communication among participants and raise awareness that circuit operations are erroneous. It could also support the comprehension of quantum noise besides the mathematical formalism of quantum channels and enhance effective quantum circuit design.

The ultimate goal would be a complete model of all cause-effect relations in a quantum computer coupled with a mathematical description of their impact on quantum circuits. Various methods exist to model such cause-effect relations. In [Weber, Riebisch, et al., 2021], we considered one of the most basic methods: Ishikawa diagrams [Ishikawa, 1976], also called fishbone diagrams. They represent a collection of all potential causes that lead to an error, but they cannot faithfully represent complex processes like quantum noise. Pearl, 2000 presents much more sophisticated models for causal relations. However, we can rarely construct a valuable model of all the causes and effects in a quantum computer because we cannot access the relevant data. Therefore, we develop an effective model for noisy processes on a quantum computer that describes only the impact on computations and not the complete cause-effect relations.

This chapter forms the foundation for such an effective description by introducing a graphical model for noise. It addresses the following overarching research question regarding the visualisation of noise models.

RQ1: How can noise models and their impact on quantum circuits be visualised?

Quantum computation without noise is commonly modelled using quantum circuits. Since their graphical representation is similar to logic circuits from classical computation, quantum circuits are intuitive to read and understand. Therefore, we consider it desirable to build on this established model and find a way to extend it to describe quantum noise. Including the noise directly in the quantum circuits shows comprehensively how they are impacted and which operations are affected. Another crucial property of quantum circuits is that quantum computers and simulators can execute them without much additional processing. Any extension of the quantum circuit model should preserve this property in order for the model to remain as practical as it currently is. Hence, we further divide our research question RQ1 into the following two subquestions.

RQ1.1: How can the quantum circuit model be extended to include noise channels?

RQ1.2: To what extent can the extended circuit model generate noisy simulations?

Our approach to generalising quantum circuits should provide practical benefits for its users. Therefore, we pursue a list of goals defined below and summarised in Table 5.1. We use these goals later as orientation for evaluating the approach.

Goal 1: Improve the communication about quantum noise between participants.

Noise models can be viewed as collections of quantum channels, i.e., mathematical equations describing the complex effects on density operators. Using such equations as a means of communication requires expertise in quantum noise, and especially interdisciplinary groups can experience challenges communicating noise models. However, quantum errors are inevitable, and everyone must face the challenge of mitigating them for meaningful quantum computation. Our approach should, therefore, support the communication about quantum noise.

Goal 2: Create a better understanding of and raise awareness for quantum noise.

The challenges for understanding quantum noise are similar to the ones for communication from above. Understanding a subject and being able to communicate it to others can be closely related. Additionally, the shortage of quantum computers makes their use often expensive, and users frequently switch to simulations. Since many simulators do not consider noise, not everybody is aware of the typical errors in a quantum device. While some errors have intuitive names that indicate which operations are affected, others can be more cryptic to unfamiliar participants (for example, thermal relaxation). We want our approach to help non-expert users understand quantum noise better and raise their awareness of errors.

Goal 3: Enable noise model simulations generated by noisy quantum circuits.

Quantum circuits are more than visualisations of programs; they are complete models of quantum computations. As such, quantum hardware or simulators can process and execute them. Any extension of the circuit model should preserve this property, at least for simulations. Depending on the modelled operations, running them on a quantum device might be impossible due to the lack of physical implementation by the device. However, our approach should be able to generate noisy simulations.

5.2. Noisy Quantum Circuits

This section presents our approach to extending the quantum circuit model with general quantum channels. We start by reflecting on the definition of quantum circuits as proposed by Deutsch, 1989, highlighting essential aspects for practical applications. Afterwards, we examine the generalisation of quantum circuits to density operators introduced by Aharonov, Kitaev, et al., 1998. We discuss their graphical representation and the usefulness of the approach. Finally, we present our definition of NQCs, including the new

#	Goal
1	Improve communication
2	Better understanding and awareness
3	Generate simulations

Table 5.1.: Goals of the noisy quantum circuit approach.

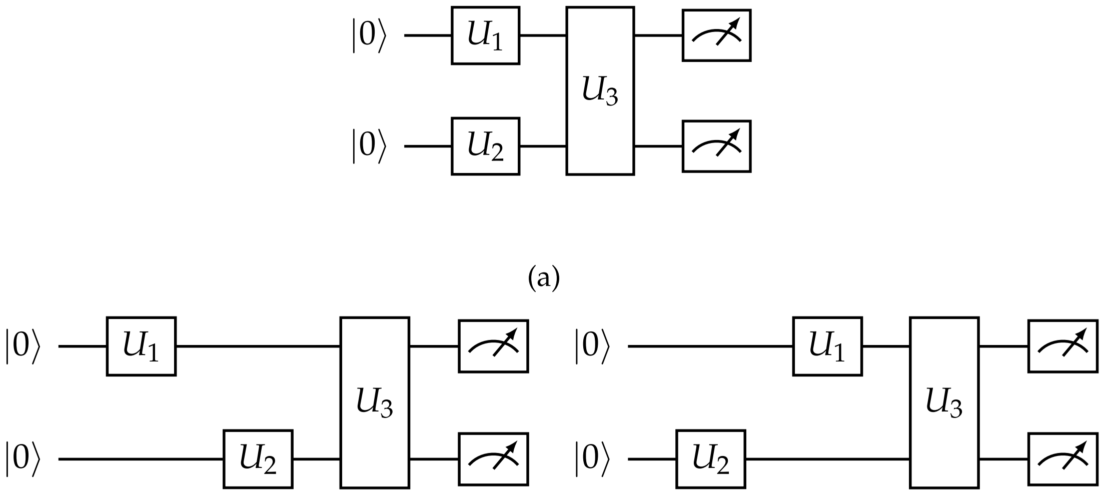
symbols for quantum channels and emphasising differences from the approach above.

5.2.1. The Quantum Circuit Model

Among many alternative models of quantum computation, the quantum circuit model has emerged as the most widely used. Deutsch, 1989 introduced it under the name "quantum computational networks" as a generalisation of the logic circuit model for classical computation. While these networks in their original form are rather theoretical, we recall quantum circuits from a more practical point of view. For example, Deutsch never considered measurements at the end of quantum circuits, but in practice, these measurements are the only way to extract information from any quantum computation. Hence, our formal definition of quantum circuits deviates from the original version for practical reasons. It describes circuits in terms of directed acyclic graphs (DAGs). A graph G consists of a finite set $V(G)$ of vertices and a finite set $E(G)$ of edges connecting two vertices. It is called *directed* if every edge has an orientation, and a directed graph is *acyclic* if it contains no loops. A vertex $v \in V(G)$ is called sink (source) if it has no outgoing (incoming) edges. See [Bang-Jensen and Gutin, 2009] for further information on DAGs. We define quantum circuits as follows:

Definition 5.2.1 (Quantum Circuit)**.** A quantum circuit on N qubits is a DAG with N sources labelled by the state $|0\rangle$ and N sinks labelled by measurements in the computational basis $\{|0\rangle, |1\rangle\}$. Every other vertex v_j is labelled by a unitary operation U_j, and the number of incoming edges of U_j coincides with the number of outgoing edges for all j.

Alternative definitions do not include the initial states and measurements in the graph but only consist of gate operations, resulting in an overall unitary matrix. However, it is more convenient for this thesis to model the complete computational process as a single graph. To execute a quantum circuit as in the definition above, we must choose a topological ordering of the gates to determine the order in which they are applied. A topological ordering means that we order the indices $j = 1, \ldots, m$ of the gates U_j such that for each directed edge from U_j to U_k, the relation $j < k$ holds. For every DAG, at least one topological ordering exists [Bang-Jensen and Gutin, 2009].



(a)

(b)

(c)

Figure 5.1.: A quantum circuit with (a) no topological ordering, (b) ordering $1 < 2 < 3$, and (c) ordering $2 < 1 < 3$.

For example, consider the circuit from Figure 5.1a, where we assume that every edge is directed from left to right. Since we have two directed edges from U_1 to U_3 and U_2 to U_3, the two-qubit gate must be applied after the single-qubit ones in every topological ordering. There are two possible choices, namely $1 < 2 < 3$ and $2 < 1 < 3$, as shown in Figure 5.1b and 5.1c. However, one can show that both choices yield the same quantum computation. Denoting the identity matrix on qubit j by I_j, first applying U_1 to the first qubit and then U_2 to the second qubit results in the unitary operation $(I_1 \otimes U_2) \circ (U_1 \otimes I_2)$. The other order gives $(U_1 \otimes I_2) \circ (I_1 \otimes U_2)$, but we can calculate

$$(I_1 \otimes U_2) \circ (U_1 \otimes I_2) = U_1 \otimes U_2 = (U_1 \otimes I_2) \circ (I_1 \otimes U_2) \,. \tag{5.1}$$

We can generalise this to arbitrary numbers of qubits. In fact, all topological orderings are computationally equivalent, and we omit this step for the remainder of the thesis. Instead, we draw quantum circuits with parallel operations if the affected qubits of different gates are disjoint, see Figure 5.1a.

Definition 5.2.1 formally introduces what is already discussed in Chapter 2. A quantum circuit prepares N qubits in the initial state $|0\rangle^{\otimes N}$, manipulates them with unitary gates, and performs measurements in the computational basis. While Deutsch, 1989 allows for qubits not to be measured at the end of a circuit, Nielsen and Chuang, 2010 show that the outcome probability distribution of the qubits with measurement is not affected by any additional qubit measurements. Hence, we assume that every circuit ends with measuring all qubits. If one is only interested in a subset of qubits, the irrelevant parts of the results can still be discarded after the measurement.

Graphically, quantum circuits are depicted by horizontal wires representing qubits, boxes or special symbols for quantum gates, and meter symbols for the measurements. They are read from left to right, and the wires are labelled with the initial state on the left side and a measurement symbol on the right. The ordering of gates transfers to their physical implementation by a quantum computer, and a piece of wire without a gate represents the idle operation that preserves the qubit state.

5.2.2. Generalisation to Density Operators

In the following, we summarise the work of Aharonov, Kitaev, et al., 1998 on generalising the circuit model to density operators. In the quantum circuit model, one can assign the qubits a well-defined state vector $|\psi\rangle$ at each point in a circuit, and every operation acting on the qubits is unitary, except for the measurement. However, modelling more general processes than quantum gates is often feasible. The authors propose an alternative definition of quantum circuits where state vectors are replaced by density matrices and quantum gates by quantum channels.

Before discussing the approach in detail, we briefly recall the most relevant definitions regarding density operators and quantum channels, see Chapter 2 for more details. Since we are interested in quantum computing using a finite number N of qubits, assume that all Hilbert spaces are finite-dimensional. A density operator or density matrix is a positive Hermitian operator $\rho \in \mathcal{B}(\mathcal{H})$ with trace equal to one. If a qubit is prepared in a pure state $|\psi\rangle \in \mathcal{H}$, the corresponding density matrix is $\rho = |\psi\rangle\langle\psi|$. A quantum channel between two systems with Hilbert spaces $\mathcal{H}$ and $\mathcal{H}'$ is a CPTP map $\mathcal{E}\colon \mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{H}')$. We restrict this to the case $\mathcal{H} = \mathcal{H}'$ later, but the approach from Aharonov, Kitaev, et al., 1998 explicitly allows different numbers of qubits before and after applying a quantum channel.

Using the above notation, they define a quantum circuit as a DAG with vertices labelled by quantum channels $\mathcal{E}_j$. Since composing quantum channels $\mathcal{E}_j$ yields another quantum channel, a quantum circuit in this approach is an overall quantum channel $\mathcal{E}$. A quantum computation is then performed by preparing qubits in an initial density matrix ρ_0, applying the quantum channel $\mathcal{E}$, and measuring a subset of the qubits. The authors present a procedure to simulate such a generalised quantum circuit by running ordinary quantum circuits on a quantum computer, using only unitary quantum gates and a set of ancilla qubits.

Our approach deviates from the one above in its definition of quantum circuits and provides a graphical representation.

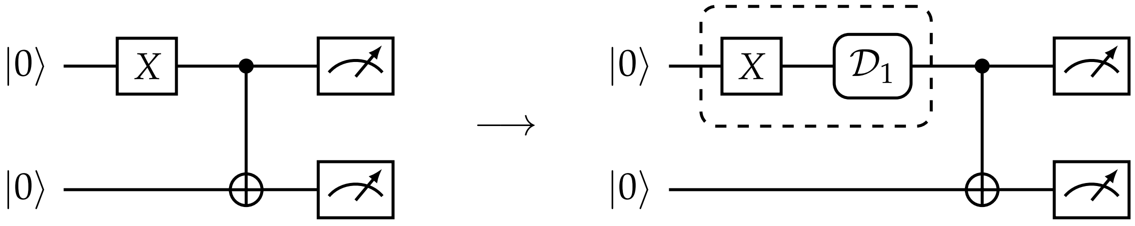


Figure 5.2.: Noise-free quantum circuit vs. same circuit with noisy X gate.

5.2.3. Noisy Quantum Circuits

We introduce our extension of the quantum circuit model with an example. The left-hand diagram in Figure 5.2 shows the noise-free quantum circuit from Section 2.1.3. After preparing two qubits in the $|0\rangle$ state, the X gate is applied to the first qubit, followed by a CX gate connecting both qubits. The right-hand diagram shows the same quantum circuit subject to noise. Recall the one-qubit depolarising channel

$$\mathcal{D}_1(\rho) = (1-\lambda)\rho + \frac{\lambda}{2} \cdot I$$

from Section 2.2. We can easily read from the diagram that the X gate is affected by depolarisation, and this depolarisation occurs directly after applying the gate. Before discussing the graphical model in more detail, we give the formal definition of this new type of quantum circuit.

Definition 5.2.2 (Noisy Quantum Circuit)**.** A NQC on N qubits is a DAG with N sources labelled by $|0\rangle$ and N sinks labelled by measurements in the computational basis $\{|0\rangle, |1\rangle\}$. Every other vertex v_j is labelled by a quantum channel $\mathcal{E}_j$, and the number of incoming and outgoing edges of $\mathcal{E}_j$ coincide for all j.

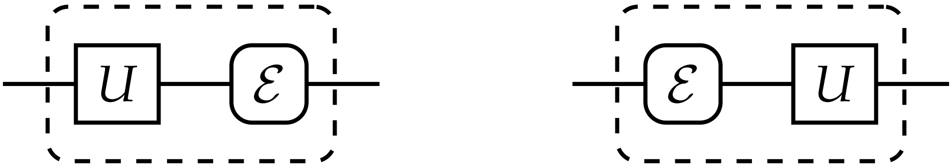
The following remarks serve the clarification of some important details. Firstly, the definition abuses the notation of state vectors and density matrices to remain as similar as possible to quantum circuits. The label $|0\rangle$ of a source refers to the density matrix $|0\rangle\langle 0|$. However, the ordinary circuit model can be restored if the circuit only consists of unitary gates. Secondly, consider a vertex with m incoming and outgoing wires labelled by a quantum channel $\mathcal{E}$. If we denote the Hilbert spaces of the corresponding qubits by $\mathcal{H}_1, \dots, \mathcal{H}_m$, then the quantum channel is a CPTP map on density operators of the composite system: $\mathcal{E}\colon \mathcal{B}(\mathcal{H}_1 \otimes \dots \otimes \mathcal{H}_m) \to \mathcal{B}(\mathcal{H}_1 \otimes \dots \otimes \mathcal{H}_m)$. Kraus' theorem states that this map $\mathcal{E}$ can be described by $2^m \times 2^m$ matrices $\{E_i\}$ with

$$\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger \, .$$

Hence, each vertex in an NQC is assigned a collection of these matrices. Note that in contrast to [Aharonov, Kitaev, et al., 1998], we aim to use generalised quantum circuits for modelling quantum noise and do not intend to run them on a quantum machine. Section 5.3 explains how NQCs relate to simulations of noise models.

5.2.4. Visualisation

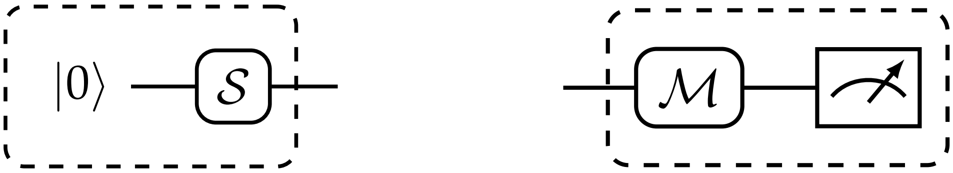
In the following, we discuss the visualisation of NQCs. Circuits with only unitary operations retain their known graphical representation. State preparation and measurement are depicted by the same symbols as before, and quantum gates are square boxes. The only two novelties are rounded boxes representing non-unitary quantum channels and additional dashed boxes indicating which operations are affected by the channel. As mentioned above, Figure 5.2 shows a first example of a noisy X gate affected by depolarising error $\mathcal{D}_1$. Noise channels can generally be modelled to occur before or after another circuit operation. Both versions of an arbitrary quantum gate U affected by a quantum channel $\mathcal{E}$ are shown in Figure 5.3.



(a) (b)

Figure 5.3.: Noisy gate U with error $\mathcal{E}$ occuring (a) after the gate, and (b) before the gate.

Throughout this thesis, we define state preparation errors to happen directly after the operation and measurement errors directly before. While it is straightforward to construct quantum channels for the former, the latter requires more effort. Chapter 7 defines measurement error in terms of Kraus matrices. Denoting state preparation error as $\mathcal{S}$ and measurement error as $\mathcal{M}$, Figure 5.4 shows the respective graphical representations, independent of the specific model.



(a) (b)

Figure 5.4.: (a) State preparation error. (b) Measurement error.

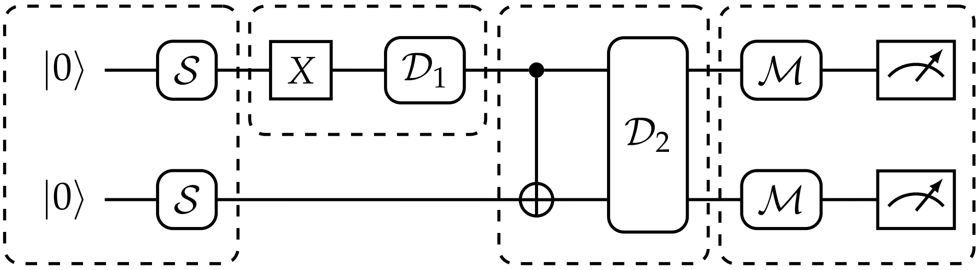


Figure 5.5.: NQC on two qubits.

For the last example in this section, we return to the two-qubit circuit from above. Assume that every circuit operation is erroneous, where state preparation and measurement are affected by the corresponding errors, and both single and two-qubit gates are subject to depolarisation. The resulting NQC is shown in Figure 5.5, where a single dashed line covers $\mathcal{S}$ and $\mathcal{M}$ for both qubits to maintain the readability.

5.3. Simulating Noise Models

Quantum circuits can directly be parsed to a quantum computer or simulator, initiating the respective machine to run the circuit and provide the results. Since we use noisy quantum circuits primarily to describe the noisy behaviour of quantum computers, we do not intend to run them on a real device because such a device is already noisy by nature. However, they should enable us to run noisy simulations to gain a real benefit. This section explains the relationship between noise models and NQCs and presents a procedure to generate simulations for them. Let **NQC** be the set of all noisy quantum circuits and $\mathbf{QC} \subset \mathbf{NQC}$ be the subset of ordinary quantum circuits. NQCs allow us to refine our definition of a noise model as follows.

Definition 5.3.1. A noise model is a map $M\colon \mathbf{QC} \to \mathbf{NQC}$ that leaves the ordering of gates in a circuit $c \in \mathbf{QC}$ unchanged but possibly adds noise channels affecting circuit operations.

Therefore, performing simulations based on a noise model is equivalent to simulating noisy quantum circuits. Recall that the latter are DAGs with vertices labelled by quantum channels (in addition to the sinks and sources). Let $c \in \mathbf{NQC}$ be an NQC with vertices $\{\mathcal{E}_j\}_{j=1,\ldots,m}$. Assume that $1 < \ldots < m$ is a topological ordering of the quantum channels (such an ordering exists because c is a DAG). Then we can simulate the noisy circuit as follows:

1. Prepare the initial density matrix $\rho_0 = |0\ldots0\rangle\langle0\ldots0|$.

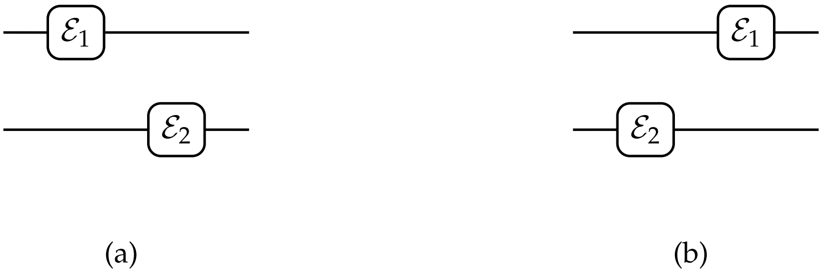


Figure 5.6.: Two quantum channels with different orderings.

2. For $j = 1, \dots, m$, apply the corresponding quantum channel $\mathcal{E}_j$ to the current density matrix: $\rho' = \mathcal{E}_j(\rho) = \sum_i E_{j,i} \rho E_{j,i}^\dagger$.

3. Apply the measurement operators to the final density matrix.

The simulation result is independent of the choice of topological ordering. Consider two quantum channels $\mathcal{E}_1$ and $\mathcal{E}_2$ applied to different qubits according to Figure 5.6. Analogously to (5.1), we can compute

$$(I_1 \otimes \mathcal{E}_2) \circ (\mathcal{E}_1 \otimes I_2) = \mathcal{E}_1 \otimes \mathcal{E}_2 = (\mathcal{E}_1 \otimes I_2) \circ (I_1 \otimes \mathcal{E}_2)\,. \tag{5.2}$$

This equation shows that not only is the computation independent of the choice of order, but we can apply two quantum channels $\mathcal{E}_1, \mathcal{E}_2$ simultaneously as the composite channel $\mathcal{E}_1 \otimes \mathcal{E}_2$. Since we only use NQCs for simulation purposes, the simultaneous execution of quantum channels does not rely on possible hardware implementations for such a composite channel.

Due to Kraus' theorem, applying a quantum channel $\mathcal{E}$ yields multiplying the density matrix with the Kraus operators E_i, see Theorem 2.2.3. The above procedure is the simplest and perhaps most natural way to obtain the outcome statistics of noisy quantum circuits and to simulate a noise model. However, various implementations of noisy simulations exist, and they do not always rely on density operators. Alternatives include probabilistic executions of unitary quantum circuits depending on the noise model. Moreover, the outcome probability distribution is not always the quantity of interest. Many quantum algorithms compute expectation values of quantum mechanical observables instead. While they could be obtained from the distributions, they can also be computed directly from the final density matrix ρ as

$$\langle O \rangle = \mathrm{Tr}(O\rho)\,. \tag{5.3}$$

See Section 2.2.1 for more details on expectation values. Finally, we explain how noise models are used to simulate quantum circuits. If we want to obtain the predictions of a noise model M for a quantum circuit $c \in \mathbf{QC}$, we can map c to a noisy quantum circuit $M(c) \in \mathbf{NQC}$ and use the simulation procedure from above. Throughout this thesis, we abuse the notation $M(c)$ to denote the noisy quantum circuit or the model predictions.

5.4. Evaluation

This section evaluates the NQC approach. The evaluation is divided into two parts: the visualisation and the simulation aspect of noise models. Firstly, we evaluate NQCs regarding their graphical representation in 5.4.1, using goals 1 and 2 from Table 5.1. Secondly, we assess the noisy simulation of quantum circuits considering goal 3 in 5.4.2. Finally, we discuss our evaluation and examine threats to validity in 5.4.3.

5.4.1. NQCs for Visualising Noise Models

In the following, we evaluate our noisy quantum circuit approach as a visualisation for noise models based on *usage scenarios*. Usage scenarios are a standard evaluation technique for software visualisations [Merino et al., 2018; Isenberg et al., 2013], and they describe how visualisation methods are expected to be utilised in practice [Sedlmair et al., 2012]. We define two scenarios addressing the first two goals of our visualisation approach, although they are not in one-to-one correspondence. Recall that the first two goals are creating a better understanding of and improving communication about quantum noise, see Section 5.1. After presenting the usage scenarios, we discuss whether NQCs can help achieve the goals.

Scenario 1 - An interdisciplinary team of researchers wants to solve an optimisation problem. After deciding to use a hybrid quantum algorithm such as VQE or quantum approximate optimization algorithm (QAOA) [Farhi, Goldstone, and Gutmann, 2014], they construct the necessary quantum circuits for the quantum subroutine of the algorithm and discuss potential errors and possible mitigation methods. An expert on quantum noise knows the most severe types of noise for this application and creates NQCs by adding them to the circuits. For example, the left-hand diagram of Figure 5.7 shows typical VQE circuits, while the right-hand diagram shows the same circuits with measurement errors. The team understands that they must counteract these errors and apply measurement error mitigation [Funcke, Hartung, Jansen, Kühn, Stornati, and Wang, 2022; Kwon and Bae, 2021].

Scenario 2 - A quantum computing provider offers a graphical interface for building quantum circuits. They can be sent as jobs to a quantum computer, which executes the circuits and returns the results. The provider constantly gathers information about

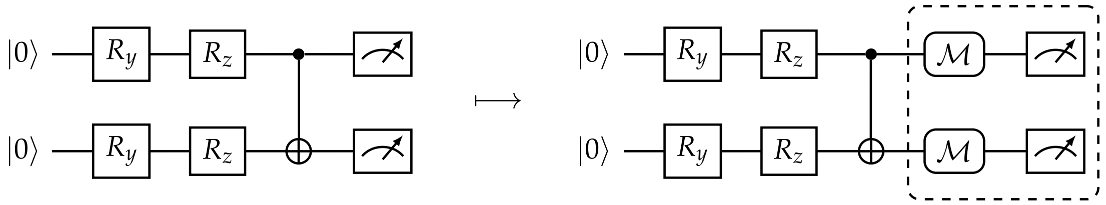


Figure 5.7.: Mapping a quantum circuit to an NQC with measurement error. The rotation angles are omitted for better readability.

quantum noise in their machines. If a user creates a quantum circuit with the interface, it automatically transforms it into an NQC containing the most dominant error types to expect in the circuit. The user is warned about potential errors and can adapt the design of their quantum circuit accordingly to minimise these errors.

Goal 1: Improve the communication about quantum noise between participants.

In the first scenario, the expert on quantum noise can explain that measurements are erroneous and that this error is the most dominant for the application of interest. No specific mathematical equations describing the errors are needed to illustrate where it occurs. Moreover, NQCs are artefacts that can be documented later, offering an advantage over alternative means of communicating noise. The second scenario shows that computer systems can also use NQCs to communicate noise models interactively.

Goal 2: Create a better understanding of and raise awareness for quantum noise.

In both scenarios, NQCs allow the users to extract valuable information about errors in their computation without requiring expert knowledge of quantum noise. Knowing the location and severity of these errors can be more important than understanding the exact processes, for example, when prioritising error mitigation methods. This data can be read from NQCs, as shown in the first scenario. Users of the interface in the second scenario are explicitly made aware of quantum noise.

5.4.2. NQCs for Noisy Simulations

Section 4.1 formulates the following goal of our approach regarding the simulation of noise models.

Goal 3: Enable noise model simulations generated by noisy quantum circuits.

Section 5.3 explains in detail how NQCs can be utilised for simulating noise models. Computing density matrices and the effects of noise channels is arguably the most intuitive way to obtain model predictions, and NQCs are the natural step from ordinary quantum circuits to noisy simulations.

5.4.3. Discussion

The evaluation with usage scenarios indicates that NQCs can help achieve goals 1 and 2 and provide a valuable tool for visualising quantum noise and noise models. Since usage scenarios generally belong to the weaker evaluation methods for visualisations [Isenberg et al., 2013], they form the largest thread to the validity of our evaluation. Case studies with experts or experiments investigating how well participants understand noise models could provide more sound evidence. However, the similarity to the widely used quantum circuit model allows users to easily familiarise themselves with the new model. NQCs can be drawn quickly and with almost no effort. Furthermore, their practical use for simulations is evident, and they can undoubtedly help to achieve goal 3.

A potential drawback of visualising quantum noise with NQCs is that diagrams can get convoluted for large and complex circuits and noise models. They should not be used to represent every quantum circuit subject to noise graphically but rather be utilised to conceptualise the errors using simple example circuits. Since their cost and learning effort are low and our evaluation shows that they can be helpful for visualisation and simulation purposes, we conclude that NQCs are a valuable tool for modelling noisy hardware behaviour. We use them to describe noise models in Chapters 6 and 7 of this thesis.

5.5. Summary

This section briefly summarises our approach to generalising quantum circuits for the visualisation of quantum noise. NQCs are quantum circuits that may contain arbitrary quantum channels instead of only unitary gates. These quantum channels are generally represented by boxes with rounded corners in the circuits. All unitary operations retain their usual symbols to preserve the ordinary quantum circuit model in the noise-free case. Additional boxes with dashed lines can be drawn to indicate which operations are affected by noise. We define NQCs formally as DAGs similar to quantum circuits but with different vertex labels. Simulations of NQCs can be derived directly from their graph representation.

CHAPTER

SIX

VOLUMETRIC BENCHMARKS FOR NOISE MODELS

Since our primary goal in this thesis is to construct a realistic quantum computing noise model, we must first be able to tell how "realistic" noise models are. Assessing and comparing them regarding accuracy is essential before we begin with their construction. This chapter presents our volumetric benchmarking approach for a systematic evaluation of noise models. It is heavily based on our publication in [Weber, Borras, et al., 2024] and is structured as follows.

Section 6.1 gives an overview of benchmarks for noise models and recollects our research questions. In Section 6.2, we introduce quality attributes for noise model benchmarks. Section 6.3 presents the volumetric benchmarking approach and explains how noise model predictions can be compared to hardware data. Finally, we evaluate this approach in Section 6.4 using the quality attributes from above.

6.1. Introduction and Research Questions

This chapter aims to answer the following research question defined in Chapter 4:

RQ2: How can noise models be evaluated and compared regarding their accuracy?

Following Kounev, 2021 and Section 2.4, a benchmark is a test for evaluating and comparing systems, tools, or techniques with respect to specific characteristics. Thus, benchmarks could offer a tailored method to investigate RQ2. While much progress has been made in testing quantum devices regarding their fidelity, only a few works deal explicitly with evaluating noise models. Since there is no standard benchmarking technique or alternative systematic evaluation approach yet, we propose a volumetric benchmarking framework to fill this gap. It compares noise model predictions to hardware data for quantum circuits of varying sizes. The circuits are typical circuits for applications of interest, such as

VQAs. It is essential to ensure high quality in terms of established quality criteria to make the benchmarks as helpful as possible. Therefore, our research focuses on the following subquestions of RQ2.

RQ2.1: Which benchmarking framework can provide valuable information on the accuracy of noise models for specific applications?

RQ2.2: How can the benchmark quality in terms of established quality attributes be ensured?

6.2. Benchmark Definition and Quality Attributes

Many criteria to assess the quality of benchmarks have been discussed for classical systems [Kistowski et al., 2015; Huppler, 2009; Skadron et al., 2003]. Due to the fundamentally different nature of quantum computers, results and standards from the classical benchmarking literature must be modified. This section adapts the definition of benchmarks and established quality attributes to our situation of evaluating quantum computing noise models.

6.2.1. Noise Model Benchmarks

Numerous similar definitions of benchmarks are formulated in the literature for different types of systems under test (for example, CPU performance). Following Sim et al., 2003 and Kistowski et al., 2015, we define benchmarks as tests for evaluating and comparing systems, tools, or techniques with respect to specific characteristics. This definition also covers the CPU benchmarks from above: They are tests to competitively evaluate and compare processors regarding their performance, energy consumption, or other properties.

Our focus is the question whether a given noise model is realistic, i.e., whether it behaves similarly to a real quantum computer. More precisely, we want to know how accurately a model can predict the erroneous hardware outcomes of quantum circuits, leading us to the following definition of noise model benchmarks.

Definition 6.2.1 (Noise Model Benchmark)**.** A noise model benchmark is a test for evaluating and comparing quantum computing noise models with respect to their accuracy.

Benchmarks typically consist of a workload that the systems under test must process. Since our goal is testing noise models, this workload is a collection of quantum circuits to be simulated by the noise models.

Chapter 5 introduces NQCs as an extension of the quantum circuit model and defines noise models as maps from ordinary quantum circuits to NQCs. Although we implicitly

Attribute	Meaning
Relevance	Generalisability to real-world applications
Reproducibility	Consistency over multiple runs
Fairness	No artificial constraints
Verifiability	Ensure correct execution
Usability	Accessible and economical

Table 6.1.: Benchmarking quality attributes.

work with this meaning of noise models in this thesis, we emphasise that alternative descriptions of noise models can always be reformulated in terms of our approach, such that we do not impose any restrictions on the models. The only requirement is the ability to simulate quantum circuits.

6.2.2. Quality Attributes

The goal of a benchmark is to give its users information on how well the systems satisfy the tested characteristics. But benchmark results initially only state how well the systems have processed the workload. Ideally, they can be generalised to real applications, but this is not always true. Similarly, there are other desirable properties of benchmarks, and like for their definitions, the terminology used for these properties can vary depending on the authors and their views. Kistowski et al., 2015 and Kounev, 2021 summarise them into five groups of quality attributes. In the following, we present the attributes and discuss them in the context of noise model benchmarks. They are used later to evaluate the volumetric benchmarking approach. Table 6.1 shows an overview of these attributes and their meaning.

- **Relevance:** The relevance of a benchmark measures the generalisability of its results to real-world scenarios of interest. For example, one might be interested in the ability of noise models to describe noisy quantum computers executing typical parametrised circuits for variational quantum algorithms. If a noise model achieves good results in a benchmark constructed for this application, its predictions should be accurate for similar circuits outside the benchmark. Therefore, a relevant benchmark requires the design of a workload capable of capturing the appropriate aspects of noise model predictions. Relevance is strongly connected to the intended use of a benchmark and can vary with different user perspectives.

- **Reproducibility:** Reproducibility refers to the consistency of benchmark results over multiple runs. Ideally, they only depend on the hardware and software environment such that running benchmarks at different times or locations (given the same environment) yields the same results. The probabilistic nature of quantum computers impedes the challenge of building reproducible benchmarks in the quantum world. This situation is aggravated for noise models due to a time drift of error rates that typically occurs on quantum devices [Fogarty et al., 2015; Klimov et al., 2018]. Another aspect of reproducibility is the documentation of a benchmark. If a benchmark involves running quantum circuits on simulators or quantum hardware, exact transpilation rules must be defined to ensure that the same transpiled circuits are executed. Other users should be able to recreate the test environment and obtain identical outcomes.

- **Fairness:** A benchmark is fair if it does not impose artificial constraints on the noise models under test. It should be unbiased, and no model should be favoured over another. Consider a benchmark that compares the predictions of noise models to hardware data. The same data set should be used for every model because the quantum device can show varying behaviour at different times, making accurate predictions more or less difficult. Another potential flaw regarding the fairness of the above benchmark could be the choice of quantum circuits. They could possess properties or symmetries that favour some models over others. A basic example is a circuit without two-qubit gates, where a well-known weakness of current quantum computers is neglected. Fairness also implies portability; performing benchmarks for various noise models should not require much implementation effort.

- **Verifiability:** There should be a high degree of confidence that a benchmark measures what it claims to measure. All of its rules must be followed when noise models simulate quantum circuits of the workload. Documenting the transpiled quantum circuits can ensure that the transpilation rules are obeyed. Moreover, presenting more data in benchmark results than is strictly needed can further enhance the trustworthiness of these results.

- **Usability:** Benchmarks should be easy to run by a user and require little expertise. They should provide precise documentation and be understandable. Moreover, a benchmark should be economical, i.e., affordable to run. Since quantum computing resources are expensive, quantum hardware data could be made available to improve accessibility.

A property that substantially impacts multiple attributes is scalability. Some authors define scalability as an additional quality attribute [Becker et al., 2022], but we interpret it as part of the others and discuss its influence on them instead. Scalability refers to the ability to run benchmarks for varying problem sizes, scaling its workload up to large

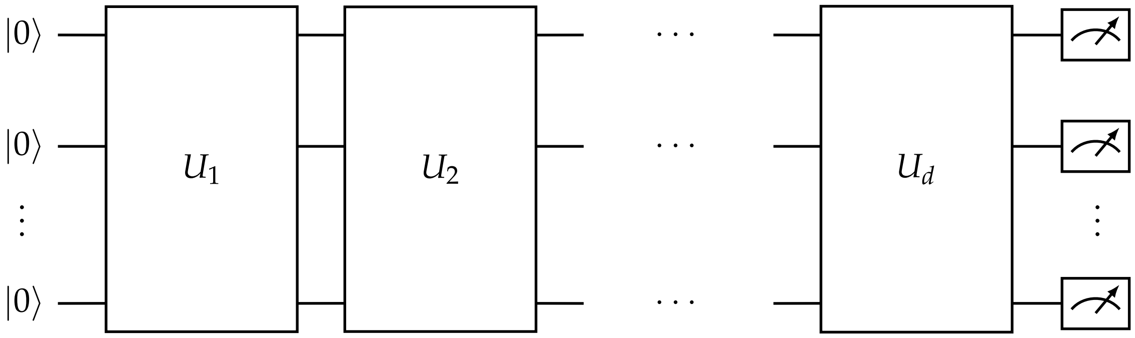


Figure 6.1.: Layered quantum circuit with d layers $U_1, \ldots, U_d$.

systems [Gray, 1994]. Since noisy simulations of quantum circuits require exponential computing resources, noise model benchmarks are inherently restricted in their scalability. Therefore, the goal is not to make them genuinely scalable but as scalable as possible.

Scalability is primarily related to three of the quality attributes: relevance, reproducibility, and usability. A scalable benchmark is more relevant because its results can more confidently be extrapolated to real-world scenarios. The exponentially growing resource requirements affect both reproducibility and usability because running a benchmark on different platforms might lead to unfeasible computing times. This applies to reproducing results in varying environments as well as the general ability to perform the benchmark.

It is generally impossible to completely satisfy all five quality attributes of a benchmark, and prioritisation is often necessary [Huppler, 2009].

6.3. Volumetric Benchmarks

This section presents our volumetric benchmark approach to evaluate noise models regarding their accuracy. It is based on the volumetric hardware benchmarks introduced by Blume-Kohout and Young, 2020. Roughly speaking, a volumetric benchmark compares model predictions to hardware results with respect to an objective function using a collection of test circuits.

This section is structured as follows. Firstly, we briefly introduce layered quantum circuits, a concept later used for the benchmarks. Secondly, we explain the benchmarking procedure in detail. Afterwards, we discuss possible choices for quantum circuits and objective functions. Finally, we demonstrate our approach by performing a volumetric benchmark for two example noise models.

6.3.1. Layered Quantum Circuits

In the following, we define layered quantum circuits and consider examples of these circuits. A quantum circuit with d layers consists of d consecutive unitary subroutines

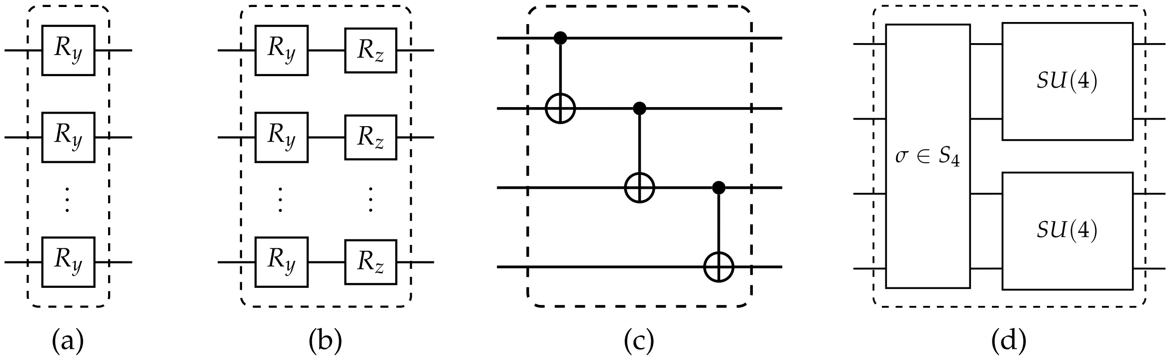


Figure 6.2.: Gate layers consisting of (a) a single R_y gate on each qubit, (b) R_y and R_z gates on each qubit, (c) a linear arrangement of CX gates, and (d) a single layer of a quantum volume circuit.

$U_1, \ldots, U_d$. The number d is called the *depth* of the circuit. For example, Figure 6.1 shows a layered circuit with depth d and subroutines that are not specified further. Each U_j consists of a collection of quantum gates, and different subroutines do not need to be built from the same gates. Possible choices of gate layers are presented in Figure 6.2. They are typical subroutines for relevant quantum computing applications. The rotational layers in 6.2a and 6.2b and the entanglement layer in 6.2c are commonly used for variational quantum algorithms (VQAs), while the subroutine in 6.2d is a part of the procedure to compute the quantum volume metric of a quantum computer [Cross, Bishop, Sheldon, et al., 2019].

Every quantum circuit can be interpreted as a layered circuit by summarising its gates into subroutines. A simple example is defining every gate as a subroutine on its own. Different groupings of gates into subroutines yield the same quantum computation but potentially varying definitions of the circuit depth.

6.3.2. Benchmark Procedure

Volumetric noise model benchmarks consist of six steps discussed in the following and visualised in Figure 6.3. Let M be the noise model to be tested and D a quantum device with native gate set $\mathcal{G}(D)$ that the model M intends to describe.

1. **Benchmark circuits:** For pairs of width w and depth d, define a set $\tilde{C}(w, d)$ of layered benchmark quantum circuits. The width is the qubit number of the circuits, while the depth refers to the number of gate layers (see Section 6.3.1). Each pair (w, d) defines a volume, hence the name volumetric benchmark. The benchmark circuits are later run by the quantum computer D and simulated by the noise model M to compare model predictions to hardware results. Suitable choices are discussed in Section 6.3.3.

2. **Circuit transpiling:** Set up rules for transpiling the quantum circuits from step 1 to

the native gate set $\mathcal{G}(D)$ of the device, enabling it to run the circuits later. There are different methods to transpile quantum circuits. For example, they can be optimised by minimising the number of physical gates applied to reduce potential gate errors in applications. While such an optimisation is often feasible for more reliable computations, it can interfere with the definition of the depth because multiple quantum gates are efficiently summarised, resulting in a shorter circuit. We denote the transpiled circuits as $C(w, d)$.

3. **Noise model predictions:** Simulate the transpiled quantum circuits $c \in C(w, d)$ for all volumes (w, d) using the benchmark noise model. We emphasise that the transpiled circuits are used for the simulation and not only for hardware experiments to ensure a fair comparison. As discussed in Section 2.1.4, simulations can be done with exact computations based on density matrices or shot-wise simulations, emulating the behaviour of a real quantum computer. The simulation process should be specified and documented in detail.

4. **Hardware experiments:** Run the transpiled quantum circuits $c \in C(w, d)$ on the quantum device. Similarly to the simulations from step 3, the exact specifications of this run should be described in detail. This includes the number of shots and the order in which the circuits are run. Moreover, the exact times of the hardware experiments should be documented.

5. **Single circuit evaluation:** Define an objective function f measuring the distance between model prediction $M(c)$ and hardware results $D(c)$ for a single quantum circuit c, and evaluate this function for all quantum circuits, i.e., compute the distance $f(M(c), D(c))$. We discuss reasonable choices for f in Section 6.3.3.

6. **Ensemble evaluation:** For each volume (w, d), compute a single number metric of the model predictions based on the outcomes of the objective function f for the individual circuits $c \in C(w, d)$. We write this final distance as $f(w, d)$. Again, we refer to Section 6.3.3, where we discuss possible options for obtaining the total assessment.

Note that we write the model predictions in step 4 as $M(c)$, which in Chapter 5 denotes the noisy circuit that M maps c to. We abuse this notation here and also use it for the simulation result of this noisy quantum circuit.

6.3.3. Benchmark Circuits and Objective Functions

Volumetric noise model benchmarks require choices for benchmark circuits and objective functions to compare model predictions to hardware results. In the following, we discuss suitable options, particularly regarding the quality attributes from Section 6.2.

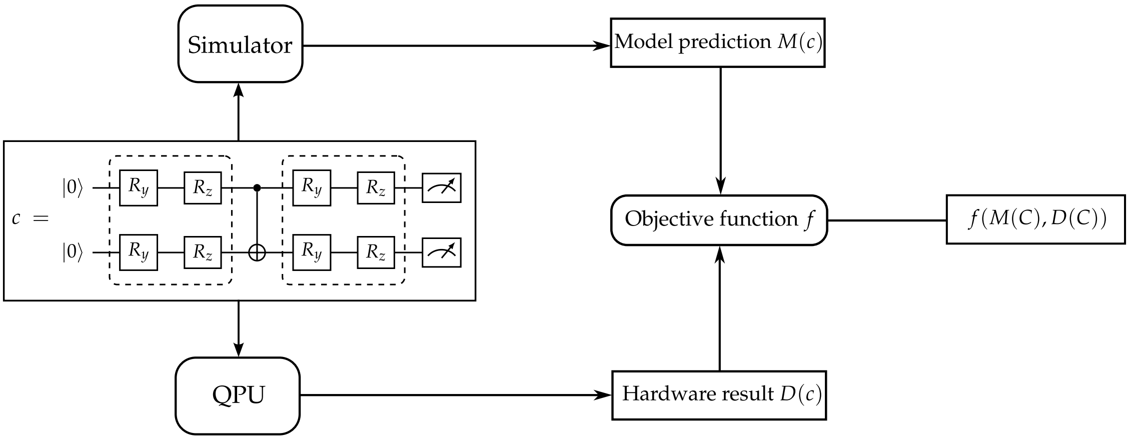


Figure 6.3.: Evaluation of objective function f for a layered quantum circuit c. The hardware is denoted by QPU (quantum processing unit).

Circuits

The first step of a volumetric benchmark is the definition of quantum circuits for different volumes (w, d). We discuss different types of layered quantum circuits in the following. Since our primary motivation for building and evaluating noise models are quantum error mitigation and realistic noisy simulations, it is intuitive to consider application-oriented quantum circuits. Alternatives include layered circuits with randomised or periodic subroutines. Blume-Kohout and Young, 2020 provide a similar discussion regarding their volumetric hardware benchmarks that we follow.

- **Application circuits:** Numerous NISQ applications implement quantum circuits with a natural layer structure. VQAs typically use alternating rotation and entanglement layers (see Figure 6.2), the QAOA applies consecutive Hamiltonian operators [Farhi, Goldstone, and Gutmann, 2014], Grover's search algorithm relies on so-called oracles and diffusers [Grover, 1996], and Quantum Neural Networks can also be built layer-wise, similarly to classical neural networks [Silva et al., 2016; Beer et al., 2020].

  One cannot always build quantum circuits of arbitrary volume (w, d) for all quantum computing applications. Some algorithms only use circuits with specific depths or widths [Blume-Kohout and Young, 2020]. However, volumetric noise model benchmarks do not intend to run the entire routine and, therefore, do not need to yield functioning computations. It is sufficient to build similar quantum circuits that reflect the general structure of the application. In this way, one can construct benchmark circuits of arbitrary volume for many relevant applications.

- **Random circuits:** Another class of benchmark circuits are random circuits. They use a pool of possible subroutines, and a circuit of depth d is constructed by randomly drawing subroutines $U_1, \ldots, U_d$ from that pool. Various quantum hardware benchmarking protocols implement random quantum circuits, including IBM's quantum volume metric [Cross, Bishop, Sheldon, et al., 2019], cross-entropy benchmarking [Boixo et al., 2018], and numerous variants of randomised benchmarking [Emerson et al., 2007], such as direct randomised benchmarking [Proctor, Carignan-Dugas, et al., 2019], cycle benchmarking [Erhard et al., 2019], or simultaneous randomised benchmarking [Gambetta et al., 2012]. Randomised compiling [Wallman and Emerson, 2016] is another algorithm relying on random quantum circuits, where Pauli gates are randomly inserted.

- **Structured circuits:** The last type of benchmark circuits we discuss are structured circuits. Subroutines are not drawn randomly but are composed in a pattern or according to more systematic rules. The most common examples are periodic benchmark circuits, where the same subroutines occur repeatedly. Practical instances of such circuits include certain tomography protocols such as long sequence gate set tomography [Nielsen, Gamble, et al., 2021] and robust phase estimation [Kimmel et al., 2015].

This classification of benchmark circuits is neither exhaustive nor unique; for example, variational circuits are typically also periodic. We emphasise that the first class of circuits is used for essential NISQ applications, and the ability to describe quantum noise accurately is crucial for such applications.

The intended usage of a noise model benchmark plays a significant role in its relevance. If the goal is to use an accurate noise model to simulate specific applications (for example, VQAs), it makes sense to build a benchmark with quantum circuits of these applications. The benchmark is then highly relevant for this area. Using less specialised circuits yields a lower relevance but for a broader usage field. The same applies to using the noise model knowledge to incorporate quantum error mitigation. Different mitigation methods might be the most effective for different applications, and benchmarking the noise model with application circuits yields the highest relevance. Since we do not yet know how much the accuracy of a noise model varies with the circuits it predicts, we recommend considering specific applications and benchmarking the model with typical circuits.

Another critical aspect is the number of benchmark circuits. If more model predictions are evaluated per volume, the benchmark results become more consistent, increasing the relevance of the benchmark and improving its reproducibility. Since variational circuits are parametrised due to the rotation layers, one can easily generate large sets of these circuits by randomising the rotation parameters, making them excellent candidates for benchmarking noise models. However, more quantum circuits make the benchmark less economical, and a balance must be found between generalisable results and usability.

Objective function

In step 5 of the volumetric benchmarking procedure, we assess the noise model predictions $M(c)$ for a single quantum circuit $c \in C(w, d)$. They are compared to the corresponding hardware results $D(c)$ with respect to an objective function f. The hardware results contain the frequencies of the obtained computational basis states. Similarly, the noise model typically predicts the probability distribution of measured basis states or the final state of the circuit (from which the former can be derived). When we interpret the frequencies from the model prediction as empirical distribution, the task of f is to compare two probability distributions. We identify the following two types of objective functions suitable for this task.

- **Statistical distance measures:** Various functions measure the distance between two probability distributions. Prominent examples, some of which are widely used in machine learning, are the Kullback-Leibler (KL) divergence [Kullback and Leibler, 1951], the total variation distance, the Wasserstein metric [Kantorovich, 1960], and the Hellinger distance [Hellinger, 1909]. Most relevant for this thesis are the KL divergence f_{KL} and the Hellinger distance f_{HD}. If P and Q are two discrete probability distributions, they are defined as

$$f_{\mathrm{KL}}(P \parallel Q) = -\sum_i p_i \log\left(\frac{q_i}{p_i}\right) , \tag{6.1}$$

$$f_{\mathrm{HD}}(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_i (\sqrt{p_i} - \sqrt{q_i})^2} . \tag{6.2}$$

- **Higher-level quantities:** Instead of directly comparing the probability distributions from the noise model and the hardware results, one can first compute derivative quantities of interest. The examples that we discuss are motivated by quantum computing applications. Many of these applications calculate the expectation values of quantum mechanical observables. We mainly focus on the operator $Z^{\otimes w}$ in this thesis. Recall that the computational basis states have the form $|k_1 \ldots k_w\rangle$, where $k_1 \ldots k_w$ is a bitstring of length w. Every basis state with an even number of $k_i = 1$ contributes the summand 1 to the expectation value $\langle Z^{\otimes w}\rangle$, while the ones with an odd number of $k_i = 1$ contribute the summand -1. Denoting the probability of measuring $|k_1 \ldots k_w\rangle$ as $p(k_1 \ldots k_w)$, we can compute

$$\left\langle Z^{\otimes w} \right\rangle = \sum_{k_1 \ldots k_w} p(k_1 \ldots k_w) \, (-1)^{k_1 + \ldots + k_w} . \tag{6.3}$$

This can be applied to the hardware results and the model predictions. Depending on the simulation method, the latter can also be obtained from the final state ρ as $\langle Z^{\otimes w} \rangle = \mathrm{Tr}(Z^{\otimes w}\rho)$, see (2.34). Computing the expectation values of other operators can be achieved by mapping them to Pauli operators P (for example, with the Jordan-Wigner transformation [Jordan and Wigner, 1928]) and adding gates to the quantum circuit to change the computational basis accordingly. Consequently, expectation values of observables generate a broad class of objective functions inspired by NISQ applications.

Ensemble evaluation

A comparison between noise model predictions and hardware results for the complete circuit ensemble is computed in the final step of the volumetric benchmarking protocol. It is based on the objective function evaluations $f(M(c), D(c))$ from the individual quantum circuits. The natural choices for the total assessment function include the arithmetic mean

$$f(w,d) = \frac{1}{m} \sum_{c \in C(w,d)} f(M(c), D(c)) \,, \tag{6.4}$$

and the median of the single-circuit results, where m is the number of circuits in $C(w,d)$. Alternatively, one can consider the worst-case scenario and compute

$$f(w,d) = \max_{c \in C(w,d)} f(M(c), D(c)) \,. \tag{6.5}$$

Such a function can be very volatile due to outliers in the benchmarks. If one wants to reduce the influence of these outliers, the following function can be considered, which is used again later in this thesis. More details are provided in Appendix B.

$$f(w,d) = \frac{1}{2} \sum_{c \in C(w,d)} \arctan\left[f(M(c), D(c))^2 \right] \,. \tag{6.6}$$

The objective function is another primary influence factor for the relevance of a benchmark besides the quantum circuits. Depending on the quantum computing application, hardware results are processed differently. For example, the Bernstein-Varizani algorithm relies on the exact bitstring of a measurement, while VQAs computes quantum mechanical expectation values. Benchmarking a noise model with an objective function capable of detecting the essential aspects improves the relevance. If model predictions are accurate regarding the $\langle Z^{\otimes w} \rangle$ expectation value in benchmark experiments, likely, they are also accurate for actual VQA experiments.

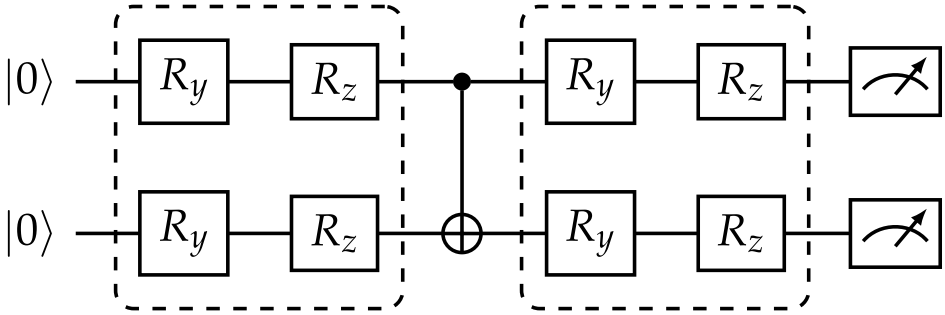


Figure 6.4.: Example of a layered circuit for $w = 2$ and $d = 3$.

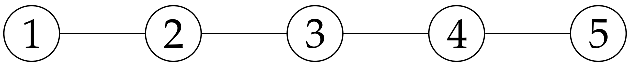


Figure 6.5.: Qubit layout of the ibmq_manila device.

6.3.4. Example

In the following, we present an instructive example of a volumetric benchmark performed with two basic models to clarify the procedure and explain the reporting of benchmark results. We start by discussing the quantum circuits we use for the benchmarks before outlining the complete process.

Benchmark circuits

We employ typical variational quantum circuits for all volumetric benchmarks performed in this thesis. They consist of alternating rotational and entanglement layers, which are shown in Figure 6.2b and 6.2c. The Qiskit library calls these circuits `EfficientSU2` and provides various options for rotation gates and entanglement structures. Here, the rotation layers consist of an R_y gate followed by an R_z gate on every qubit, while the CX gates connect the qubits linearly, beginning with the first qubit. Figure 6.4 shows an example of such a quantum circuit with $w = 2$ qubits and $d = 3$ layers, where the rotation parameters are omitted for better readability. We adapt Qiskit's terminology and refer to these benchmark circuits as `EfficientSU2`.

Benchmark procedure

In the following, we describe the volumetric benchmarks performed in this section. Since they serve as instructive examples and are not the focus of our research, we give a basic overview of the different benchmark steps and refer to Chapter 7 for more detailed documentation, including a more comprehensive discussion on the choices of circuits and objective functions.

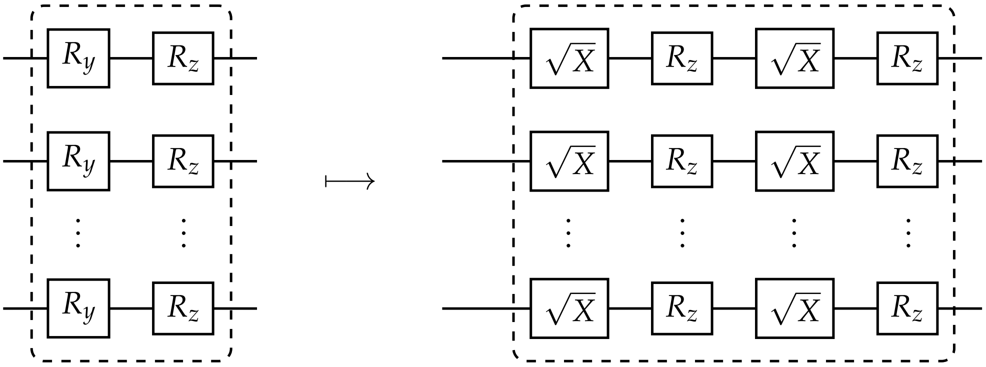


Figure 6.6.: Transpiling a rotation layer of R_y and R_z gates into the native gate set.

The quantum device D that runs the hardware experiments is IBM's ibmq_manila device with its native gate set $\mathcal{G} = \{X, \sqrt{X}, R_z, CX\}$. It consists of five qubits in a linear layout, where only neighbouring qubits are natively connected by CX gates. The layout is shown in Figure 6.5. We benchmark two noise models in this section. The first model contains no noise channels and describes a noise-free quantum computer with ideal results, while the second model considers all CX gates subject to two-qubit depolarising error $\mathcal{D}_2(\rho) = (1 - \lambda\rho) + \lambda/4 \cdot I$. The error parameter of the depolarising channels is fixed as $\lambda = 0.03$ for all gates. We denote the two models as M_{nf} and M_{depol}, and the benchmarks follow the procedure below.

1. **Benchmark circuits:** For each volume (w, d), we use a set $\tilde{C}(w, d)$ consisting of 200 `EfficientSU2` circuits, as described above. The rotation parameters are drawn uniformly from the interval $[0, 2\pi)$.

2. **Circuit transpiling:** Qiskit offers several optimisation options for the transpiling of quantum circuits. We use the option `optimization_level = 2`. Figure 6.6 shows how it transpiles a rotation layer of R_y and R_z gates into the native gates. The entanglement layers are preserved because the CX gate is already native.

3. **Noise model predictions:** We use exact density matrix computations to obtain the noise model predictions. All gate operations and noise channels are formulated as Kraus operators and applied consecutively to the initial density matrix. The expectation value $\langle Z^{\otimes w} \rangle$ is computed as $\langle Z^{\otimes w} \rangle = \text{Tr}(Z^{\otimes w}\rho)$, following (2.34).

4. **Hardware experiments:** All hardware experiments are performed using $s = 8192$ shots per quantum circuit. The exact computation times can be found in Table B.2 in Appendix B.

5. **Single circuit evaluation:** The objective function comparing model predictions to

Noise-free model M_{nf}

	$d = 1$	$d = 2$	$d = 3$	$d = 4$	$d = 5$
$w = 1$	0.0362	0.0315	0.0323	0.0307	0.0320
$w = 2$	0.0494	0.0736	0.0685	0.0718	0.0556
$w = 3$	0.0776	0.0752	0.0585	0.0694	0.0928
$w = 4$	0.0639	0.0831	0.0591	0.0524	0.0684
$w = 5$	0.0316	0.0820	0.0458	0.0455	0.0631

Depolarisation model M_{depol}

	$d = 1$	$d = 2$	$d = 3$	$d = 4$	$d = 5$
$w = 1$	0.0362	0.0315	0.0323	0.0307	0.0320
$w = 2$	0.0494	0.0552	0.0582	0.0502	0.0391
$w = 3$	0.0776	0.0530	0.0463	0.0434	0.0657
$w = 4$	0.0639	0.0514	0.0426	0.0284	0.0396
$w = 5$	0.0316	0.0568	0.0334	0.0267	0.0399

Table 6.2.: Numeric values of volumetric benchmark for noise-free model M_{nf} and depolarisation model M_{depol}.

hardware results for a single quantum circuit $c \in C(w, d)$ is the absolute difference in expectation values:

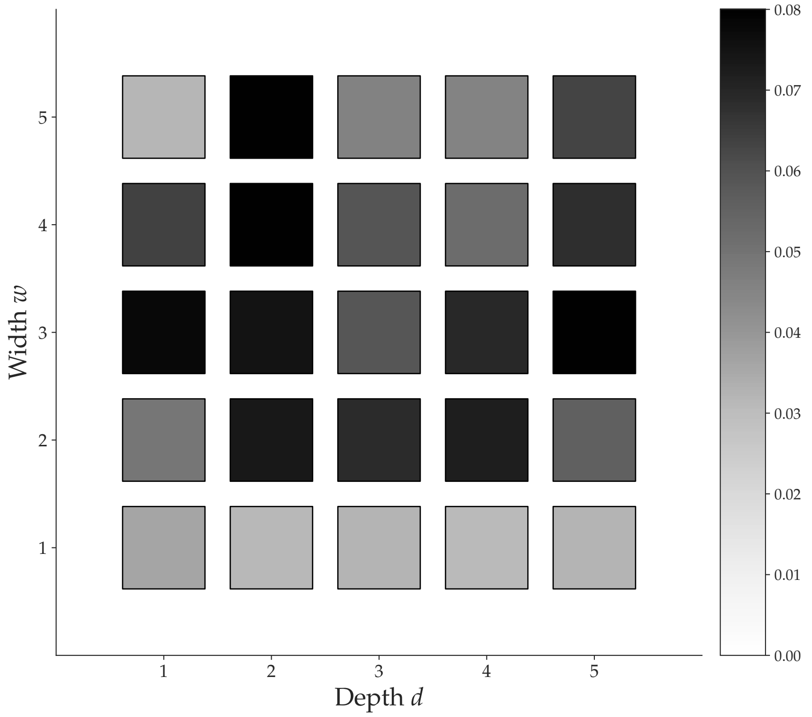
$$f(M(c), D(c)) = \left| \langle Z^{\otimes w} \rangle_{M(c)} - \langle Z^{\otimes w} \rangle_{D(c)} \right| . \tag{6.7}$$

6. **Ensemble evaluation:** The volumetric benchmark evaluation of the complete ensemble is computed as the arithmetic mean over all single-circuit results for the respective volume (w, d), see (6.4).

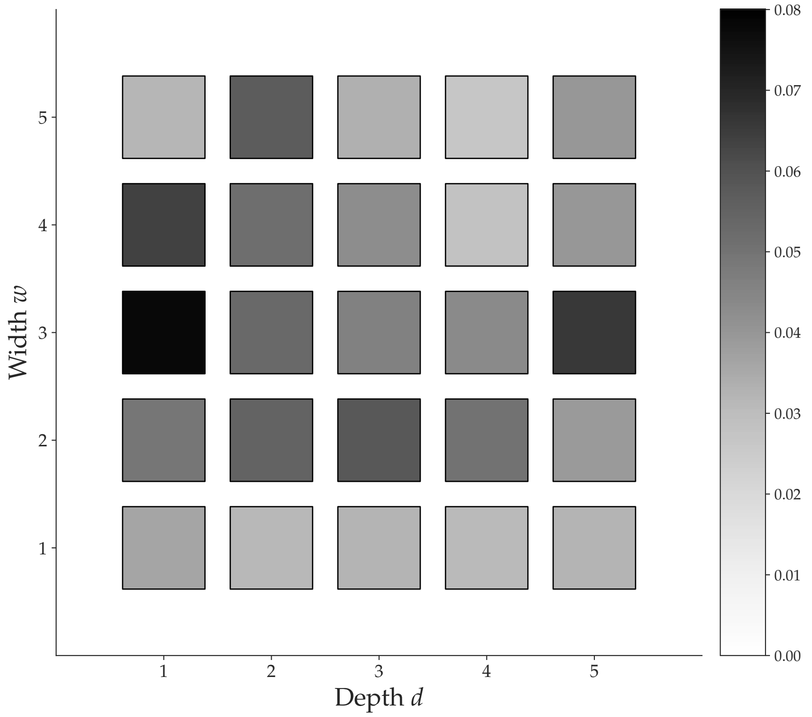
As mentioned above, more details on the volumetric benchmarks and the implementation are described in Chapter 7 and Appendix B.

Benchmark results

Figure 6.7 shows the results of the volumetric benchmarks for the noise-free model M_{nf} and the depolarising model M_{depol}. Each square in the plot represents the results of one volume (w, d). While the x-axis denotes the depth d of the quantum circuits, the y-axis describes the width w, i.e., the number of qubits. The scale on the right side indicates how the colours of the squares translate to the objective function evaluation. Brighter squares



(a) Noise-free model M_{nf}.



(b) Depolarisation model M_{depol}.

Figure 6.7.: Volumetric benchmark results for two basic noise models. The coloured scale on the right side shows the average absolute error $|\langle Z^{\otimes w} \rangle_M - \langle Z^{\otimes w} \rangle_D|$.

correspond to better model predictions and, conversely, darker squares mean less accurate predictions. Table 6.2 presents the numeric values of the benchmarks.

Since the noise-free model assumes an ideal quantum computer, the deviations between its predictions and reality are large. However, one can also interpret this result as a hardware benchmark for the ibmq_manila device. Without any quantum error mitigation, one can expect the value of $\langle Z^{\otimes w} \rangle$ to differ from the ideal one by at least 0.05 for circuits with more than a single qubit. The deviation generally has an upper bound of 2 because the expectation value $\langle Z^{\otimes w} \rangle$ is bounded by $+1$ and -1. Hence, the average error is more than 2% of its maximum possible value for many configurations. Although the depolarisation model is only a basic example, we can see a significant improvement compared to the noise-free case.

One can provide additional information, such as confidence intervals, to improve the trustworthiness of benchmarks. Figure 6.8 shows an alternative visualisation of the volumetric benchmark results. Each bar diagram represents one qubit number w, and the bars correspond to the different circuit depths d. We compute the confidence intervals using bootstrap. Again, details on the method are provided in the later chapters of this thesis because the plots shown here only serve as instructive examples.

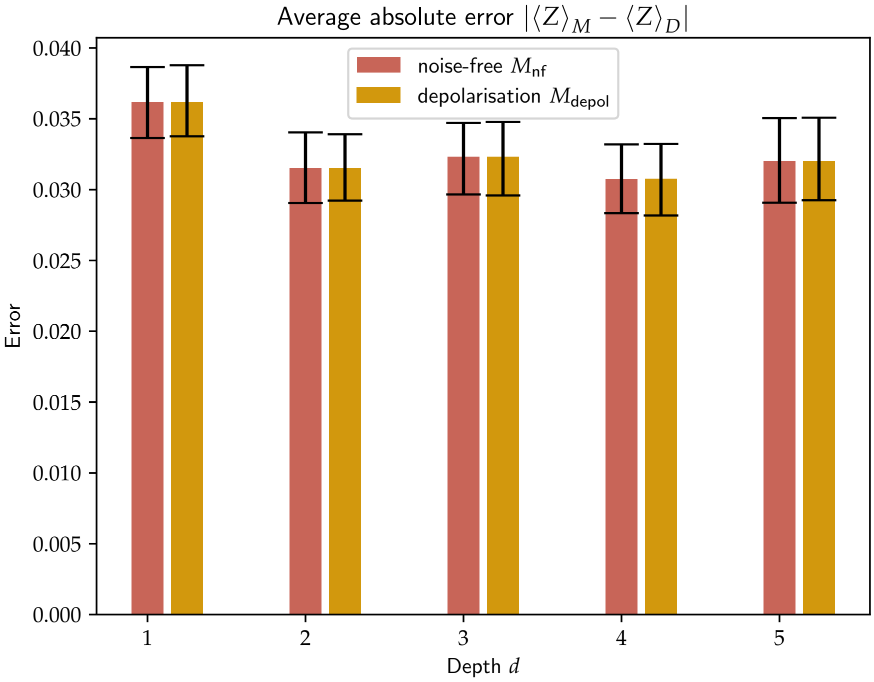
6.4. Evaluation

We conclude this chapter by evaluating the volumetric noise model benchmarks using the benchmarking quality attributes. Since we provide a benchmarking *framework* instead of concrete instances, the benchmark quality hinges on the choices for different steps of the benchmark process, including the quantum circuits and objective function. Section 6.3.3 presents suitable options and outlines their impact on quality attributes. In the following, we extend this discussion and assess the potential strengths and weaknesses of our approach. Table 6.3 provides a summary of this evaluation. We demonstrate how the framework can be used to construct high-quality benchmarks, but in the end, it is the task of the creators to ensure this quality.

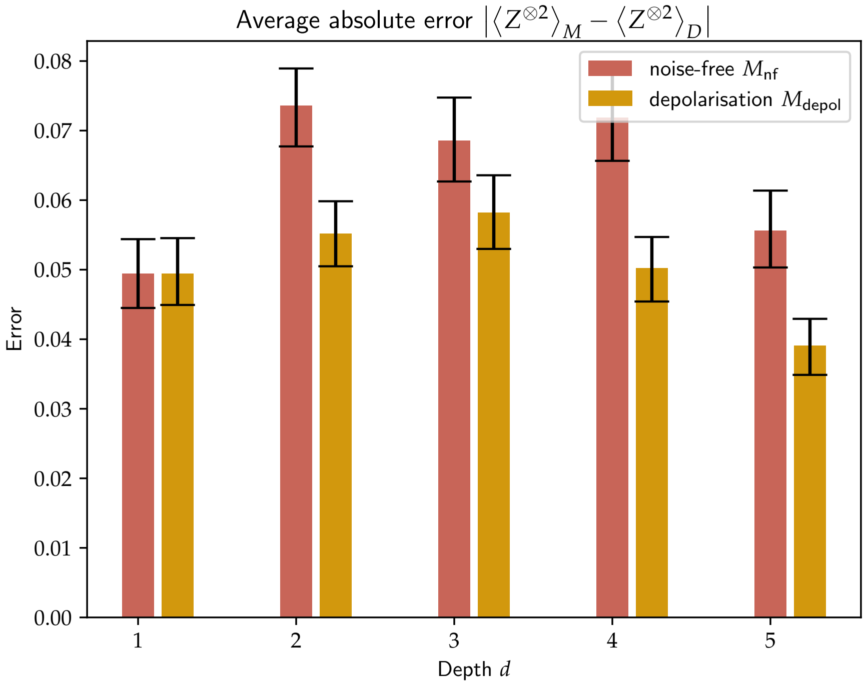
6.4.1. Quality attributes

Relevance

The relevance of a volumetric noise model benchmark strongly depends on the intended usage, the benchmark circuits, and the objective function. There is generally a trade-off between its relevance for a given area and the breadth of this area [Kounev, 2021]. Broader applicability typically decreases the relevance, while the specialisation on a narrower field increases it. Since the practical benefits of accurate noise models lie in applying quantum

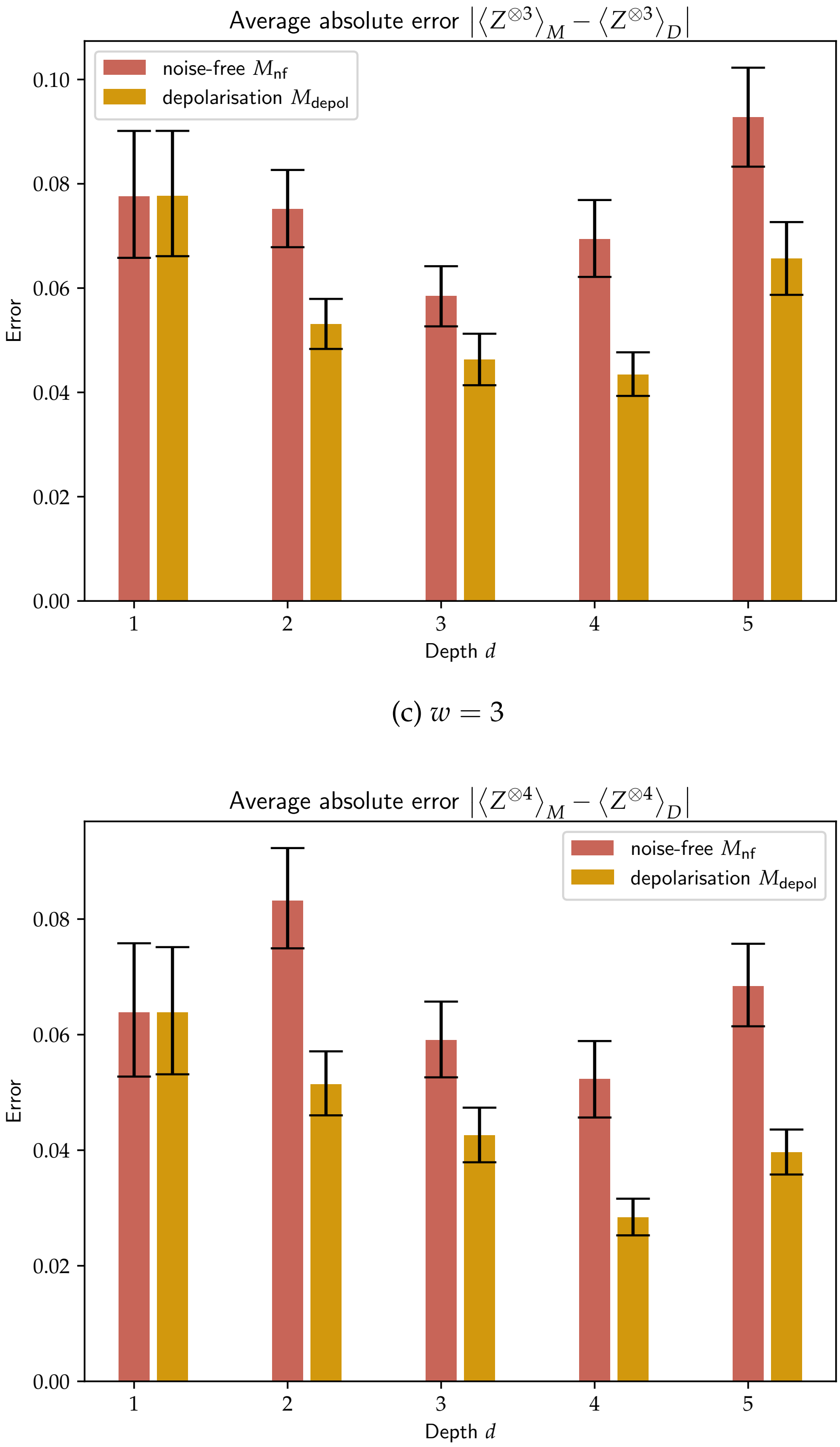


(a) $w = 1$



(b) $w = 2$

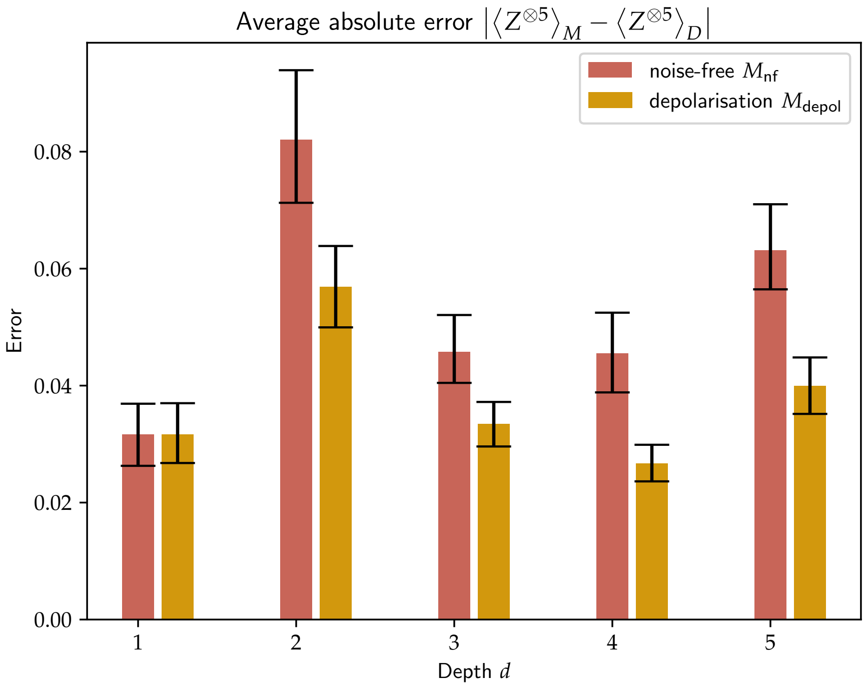
Figure 6.8.: Volumetric benchmark results for two basic noise models with error bars.



(c) $w = 3$

(d) $w = 4$

Figure 6.8.: Volumetric benchmark results for two basic noise models with error bars.



(e) $w = 5$

Figure 6.8.: Volumetric benchmark results for two basic noise models with error bars.

error mitigation and realistic simulations, we expect them to be constructed for specific areas. Therefore, our evaluation focuses on the relevance of volumetric benchmarks for narrower applicability, and we assume that the usage of the benchmarks is specified.

The quantum circuits $C(w, d)$ and objective function f of a benchmark are the essential factors influencing its relevance. Extrapolating from the performance of a noise model in a benchmark to its accuracy in real-world scenarios requires their capability to represent the essential properties of these scenarios. Every quantum computing application consists of running quantum circuits on a device and extracting information from the results. A benchmark can achieve high relevance if its circuits are similar to the application circuits, and if the objective function approximates its information extraction process.

We demonstrate this using VQAs as instances of a quantum algorithms. Suppose we seek to simulate a VQA with a realistic noise model. In that case, we should benchmark this model using variational quantum circuits and an objective function that measures the expectation values of typical Hamiltonians or their building blocks. `EfficientSU2` circuits and the absolute difference in $\langle Z^{\otimes w}\rangle$ as the objective function fulfil this criterion.

One of the main challenges concerning relevance (and other quality attributes) is the natural statistical variance of quantum experiments. Every quantum circuit is run a finite number of times, and particularly benchmarks are based on finite samples. Many benchmark circuits should be used, and large shot numbers should be chosen to mitigate this effect. Moreover, one can compute confidence intervals for the results to estimate the variance.

The other primary concern is the time drift observed in the error rates of quantum computers. A noise model benchmarked as realistic using quantum experiments performed at a specific time might not retain its accuracy for the same quantum device. For example, Burnett et al., 2019 investigate the drift of the T_1 decoherence time in superconducting qubits. T_1 is a typical parameter of thermal relaxation error. Every noise model should be tested with the same hardware data set to avoid corruption of benchmark results due to time drift. If the noise models have tunable parameters, one could merge the data from different times and adapt the model parameters for every sample accordingly.

Reproducibility

The reproducibility of volumetric noise model benchmarks is susceptible to the same main threats as their relevance: statistical fluctuations and time drift. However, the same countermeasures can be applied. Running many quantum circuits with large shot numbers improves confidence in benchmark results and increases the likelihood that repeated benchmarks yield consistent outcomes. Moreover, exact simulations of noise models based on density matrices can reduce the shot noise further. Since error rates experience drift over time, it is crucial to compare all noise models of a benchmark to the same hardware data set.

Another potential cause of inconsistencies is the transpiling process. In step 2 of the volumetric benchmark procedure, the benchmark circuits are transpiled into the native device gates. As highlighted in the corresponding section, the transpilation rules must be precisely documented. The transpiled circuits should be validated to avoid incorrect transpilation.

Fairness

Unfair advantages of noise models can arise when the choice of the objective function or benchmark circuits favours them over others. If the circuits are too simple and specific errors are inherently avoided, a benchmark cannot measure the actual accuracy of noise models. Instead, the noise model with the best description of the relevant subset of errors achieves the best results. For instance, if the circuits contain no entanglement, one of the most severe sources of errors is not taken into account. Choosing complex and expressive quantum circuits is a possible countermeasure.

A similar situation can occur regarding the objective function f. If it is too superficial, a noise model could approximate the hardware behaviour accurately with respect to f

Attribute	Threat	Countermeasure(s)
Relevance/ Reproducibility	Statistical fluctuations	Use many circuits Execute with large shot number Compute confidence intervals
	Time drift	Use same hardware data for all models Merge data from different times
	Inconsistent transpiling	Validate transpiled circuits
Fairness	f favours certain models	Test several objective functions Ensure fair circuits and function
	Circuits too simple	Use complex and expressive circuits
Verifiability	Incorrect circuit runs	Check for anomalies in data
	Erroneous simulations	Simulate simple test models Simulate basic quantum circuits Compare different simulators
	Inconsistent transpiling	Validate transpiled circuits
Usability	Restricted hardware access	Provide quantum data set

Table 6.3.: Threats to the benchmark quality and countermeasures.

without providing a realistic representation of quantum noise. This scenario can be avoided by testing multiple objective functions and choosing them to be sufficiently complex.

Verifiability

There are three steps in the benchmarking procedure where errors are likely to occur: transpilation, hardware experiments, and noisy simulation. Incorrect circuit transpiling can easily be avoided by validating the transpiled circuits. The noisy simulations can be verified using basic test models or circuits, where the expected simulation outcomes can be predicted. Moreover, different simulation tools can be compared to increase the probability of correct processing.

The hardware experiments pose a more considerable challenge for the verifiability of benchmarks. Since our goal is to investigate the noisy nature of current quantum hardware, all quantum experiments are erroneous. These errors are expected to occur to some extent, but unforeseeable events might cause unusual machine behaviour, rendering hardware results useless. Evaluating the data with a noise-free simulation can provide an estimate of the noise levels observed in the quantum device. They can be compared to regular calibration data to assess whether the noise is within a typical range for the device.

Usability

The realisation of volumetric benchmarks can be divided into the classical and the quantum part. Since the benchmark procedure is straightforward, implementing the classical part can easily be achieved given precise documentation of the individual steps. However, quantum computing resources are limited, and their access is restricted. Quantum hardware data sets should be made public for all researchers to use, particularly to perform volumetric benchmarks.

Scalability

Although we do not consider scalability a distinct benchmarking quality attribute, we discuss it separately to avoid repetitions because it affects almost all other criteria. The complexity of simulating quantum circuits, particularly with quantum noise, grows exponentially with the number of qubits. Therefore, the simulation step of volumetric benchmarks naturally restricts the maximal volume we can test. This restriction impacts relevance, reproducibility, verifiability, and usability. One cannot necessarily generalise benchmark results to larger systems, and running a benchmark with many qubits requires heavy classical computing resources (or time). However, this is not a problem specific to our approach, but it occurs in every scenario where noise models are benchmarked. A possible approach to overcome the scalability problem is provided by partial benchmarks that use subsets of qubits to compare model predictions to hardware data.

6.4.2. Discussion

Noise model benchmarks can be evaluated using established quality attributes. This evaluation is limited because we propose a benchmarking framework and not specific instances in this chapter, and the quality of volumetric benchmarks in terms of the quality attributes can vary greatly. However, we examine potential threats to every criterion and explain how they can be mitigated to improve the benchmark quality. We provide concrete choices for benchmark circuits and objective functions that yield highly relevant benchmarks for the application context of VQAs. In Chapter 7, we construct a quantum computing noise model and use our benchmarking approach to assess its accuracy compared to other noise models.

To answer the research questions formulated at the beginning of the chapter: We find that volumetric benchmarks can provide a systematic evaluation and comparison of quantum computing noise models if they are constructed according to their intended usage. Choosing quantum circuits and objective functions capable of representing this usage yields high quality in terms of the quality attributes from the literature.

CHAPTER

SEVEN

NOISE MODEL CONSTRUCTION

This chapter presents the main contribution of this thesis. It introduces a quantum computing noise model and shows how the model parameters can be optimised systematically. The resulting optimised noise model is evaluated using our volumetric benchmarking approach from Chapter 6. Parts of what we present below has been published in [Weber, Borras, et al., 2024]. However, here we provide some additional analysis of the noise model parameters and new optimisation strategies.

The chapter is structured as follows. Section 7.1 recalls the research questions investigated in this chapter. Section 7.2 presents our full noise model, and Section 7.3 explains how its parameters can be optimised. In Section 7.4, we evaluate our noise model using volumetric benchmarks. Finally, Section 7.5 summarises the research of this chapter.

7.1. Introduction and Research Questions

Accurately modelling quantum noise is crucial to understanding and predicting the noisy behaviour of current quantum hardware, allowing us to incorporate quantum error mitigation and realistically emulate quantum computers. Therefore, our primary goal in this thesis is to *construct an understandable noise model that is more realistic than existing noise models from the literature*. After introducing tools for visualising and evaluating quantum computing noise models in the previous chapters, this chapter aims to achieve its goal with a method to construct and optimise noise models built on the underlying physics processes causing errors. It pursues the following research question:

RQ3: How can realistic noise models be constructed?

Following Chapter 5, noise models assign noise channels to quantum circuit operations. They represent various errors occurring during quantum computations. For example,

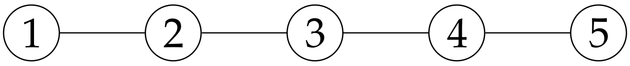


Figure 7.1.: Qubit layout of the ibmq_manila device.

the readout of qubits at the end of a quantum circuit is typically imperfect [Kwon and Bae, 2021]. Noise channels and, thereby, noise models depend on specific parameters. For instance, the abovementioned readout error can be modelled as a bit-flip affecting the measurement outcome with a certain probability. This probability is a typical noise parameter. The variety of possible errors and their dependence on specific parameters give rise to further research questions:

RQ3.1: What types of noise should an accurate noise model include?

RQ3.2: How can noise model parameters be optimised?

7.2. Full Noise Model

This section comprehensively explains our full noise model M and all errors it includes. The model generally considers all circuit operations erroneous. State preparation, quantum gates, and measurement are affected by different noise channels. These noise channels depend on parameters that we determine using an approach inspired by machine learning. The number of parameters scales linearly with the number of qubits in a linear layout. Since NISQ devices are not expected to provide full qubit connectivity, considering linear qubit layouts is a reasonable assumption. We begin this section with a brief overview of the general setting of our model constructions. Afterwards, we describe all noise channels the model assigns to quantum circuits. Finally, we present the complete composition and discuss the model parameters.

7.2.1. Setting

The three defining properties of a quantum computer regarding our construction are the number of qubits, the qubit layout, and its native gate set. While we consider a specific quantum device with its properties in this thesis, the construction can easily be adapted to other machines with different attributes.

We describe our noise model using IBM's ibmq_manila quantum computer as an orientation for the device properties. It consists of five qubits in a linear layout, as shown in Figure 7.1, and its native gate set is $\mathcal{G} = \{X, \sqrt{X}, R_z, CX\}$. Therefore, we focus on N linearly arranged qubits and assume for simplicity that CX gates are only applied in one of the two possible directions.

7.2.2. Errors

In the following, we describe the five types of errors included in our noise model. We give their representation in terms of Kraus operators because they are practically helpful for noisy simulations. Recall Kraus' theorem 2.2.3 from Section 2.2.2 stating that every noise channel $\mathcal{E}$ can be written as

$$\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger \ , \tag{7.1}$$

where $E_i \colon \mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{H})$ are linear operators such that $\sum_i E_i^\dagger E_i = I$.

State preparation error

All qubits are initially prepared in the state $|0\rangle\langle 0|$ at the beginning of every quantum circuit. This state preparation of a qubit is performed incorrectly in our noise model, and the initial state is flipped to $|1\rangle\langle 1|$ with probability p_s. We model this flip with the noise channel

$$\mathcal{S}(\rho) = (1 - p_s) \cdot \rho + p_s \cdot X \rho X \ . \tag{7.2}$$

It can alternatively be written as a combination $\mathcal{S}(\rho) = S_0 \rho S_0 + S_1 \rho S_1$ of the following Kraus operators:

$$S_0 = \sqrt{1 - p_s} \cdot I \ , \tag{7.3}$$

$$S_1 = \sqrt{p_s} \cdot X \ . \tag{7.4}$$

Our model allows for potentially different error probabilities for each qubit. We denote the probability of erroneous state preparation on qubit q by $p_s(q)$. If we write the state preparation error on qubit q as $\mathcal{S}_q$, then the overall state preparation error is the tensor product

$$\mathcal{S} = \mathcal{S}_1 \otimes \ldots \otimes \mathcal{S}_N \ , \tag{7.5}$$

affecting all qubits simultaneously. State preparation error adds one parameter to our noise model for every qubit, i.e., there are N probabilities $p_s(q)$.

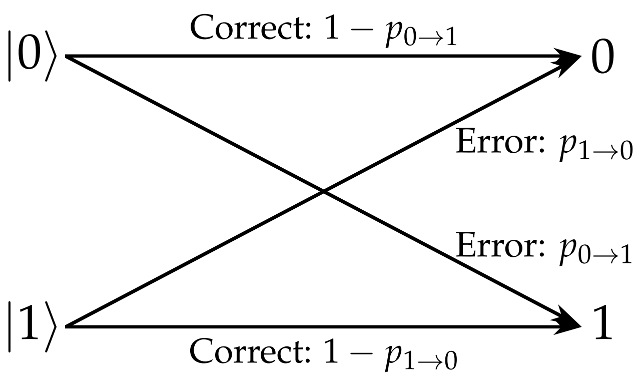


Figure 7.2.: Classical bit-flip model of measurement error.

Measurement error

Measuring qubits in their computational basis is essential to performing quantum computations. Unfortunately, it is known to be imperfect for quantum devices of the NISQ era. We model measurement error following Funcke, Hartung, Jansen, Kühn, Stornati, and Wang, 2022, where the classical bits obtained from qubit measurements are flipped. With probability $p_{0\to1}$, the incorrect state 1 is measured instead of the correct outcome 0, and vice versa with probability $p_{1\to0}$. Figure 7.2 illustrates this behaviour. The bit-flip probabilities can differ for each qubit, and we write $p_{0\to1}(q), p_{1\to0}(q)$ for the probabilities corresponding to qubit q.

While the abovementioned measurement error model considers classical bit-flips after the measurements, we present an equivalent representation in terms of Kraus operators acting directly before the measurement. To the best of our knowledge, this is the first Kraus operator representation of measurement error in the literature. It is less instructive than Figure 7.2 but more practical for simulation purposes. It only makes sense as a specific manipulation of the density matrix right before the readout. The Kraus operators for a single qubit are:

$$\begin{aligned}
M_0 &= \sqrt{(1-p_{0\to1})(1-p_{1\to0})} \cdot I\,, \\
M_1 &= \sqrt{(1-p_{0\to1})p_{1\to0}} \cdot |0\rangle\langle 0|\,, \\
M_2 &= \sqrt{(1-p_{0\to1})p_{1\to0}} \cdot |0\rangle\langle 1|\,, \\
M_3 &= \sqrt{p_{0\to1}(1-p_{1\to0})} \cdot |1\rangle\langle 0|\,, \\
M_4 &= \sqrt{p_{0\to1}(1-p_{1\to0})} \cdot |1\rangle\langle 1|\,, \\
M_5 &= \sqrt{p_{0\to1}p_{1\to0}} \cdot X\,.
\end{aligned} \tag{7.6}$$

The quantum channel for the measurement error $\mathcal{M}$ can then be written as

$$\mathcal{M}(\rho) = \sum_{i=0}^{5} M_i \rho M_i^\dagger \, . \tag{7.7}$$

This can be extended to all qubits $q = 1, \ldots, N$ in analogy to (7.5) using their corresponding bit-flip probabilities $p_{0\to1}(q), p_{1\to0}(q)$.

Gate errors

Since gate operations in quantum circuits rely on interactions of the qubits with their environment, they are generally prone to error. Our noise model includes three types of noise affecting quantum gates: crosstalk, depolarisation, thermal relaxation/dephasing, and amplitude damping. All three errors are discussed below.

- **Crosstalk:** The term crosstalk covers a wide range of processes, as discussed by Sarovar et al., 2020. In this thesis, we focus on *pulse spillover*. Pulse spillover describes the effect of quantum gates influencing neighboured qubits to the intended target qubit. Whenever an X or $\sqrt{X}$ gate is applied to a qubit, our noise model induces a small $R_x(\phi)$ rotation on its neighbours. The angle ϕ is a model parameter and can differ for each type of quantum gate $g \in \{X, \sqrt{X}\}$ and qubit q; we write $\phi_g(q)$. The corresponding noise channel is a unitary operation $\mathcal{C}(\rho) = C\rho C^\dagger$ with a single Kraus operator

  $$C = R_x(\phi_g(q)) \, . \tag{7.8}$$

  Since we consider two gates causing crosstalk error and the error is described by one rotation angle $\phi_g(q)$ for each gate and qubit, this type of noise adds $2N$ parameters to our noise model.

- **Depolarisation:** The second type of noise affecting quantum gates is depolarisation. Although Sections 2.2.2 and 5.2.3 already introduced depolarising error as an example noise channel, we discuss it again in more detail. Consider the density operator of a single qubit given by the normalised identity matrix: $\rho = I/2$. Recall from Section 2.2.1 that the trace of ρ^2 measures whether a state is mixed or pure, i.e., whether there is a state vector $|\psi\rangle \in \mathcal{H}$ with $\rho = |\psi\rangle\langle\psi|$. Computing this trace for the identity matrix from above yields

  $$\mathrm{Tr}\left(\rho^2\right) = \mathrm{Tr}\left(\frac{I}{4}\right) = \frac{1}{2} \, , \tag{7.9}$$

and we observe that ρ is a mixed state as the trace is less than one. It is indeed maximally mixed and contains minimal information about the system. Depolarisation error causes the transition of a qubit into this maximally mixed state with a specific probability λ. It acts on N qubits as

$$\mathcal{D}_N(\rho) = (1-\lambda)\rho + \frac{\lambda}{D} \cdot I \,, \tag{7.10}$$

where $D = 2^N$ is the dimension of the Hilbert space.

In the following, we present the Kraus operators describing depolarising noise on one and two qubits. The single qubit depolarising error can be written as

$$\mathcal{D}_1(\rho) = \left(1 - \frac{3\lambda}{4}\right) \cdot \rho + \lambda \cdot \frac{X\rho X + Y\rho Y + Z\rho Z}{4} \,. \tag{7.11}$$

Hence, each Pauli matrix acts with probability $\lambda/4$, and the Kraus operators are

$$\begin{aligned} D_1^{(0)} &= \sqrt{1 - \frac{3\lambda}{4}} \cdot I \,, \\ D_1^{(1)} &= \sqrt{\frac{\lambda}{4}} \cdot X \,, \\ D_1^{(2)} &= \sqrt{\frac{\lambda}{4}} \cdot Y \,, \\ D_1^{(3)} &= \sqrt{\frac{\lambda}{4}} \cdot Z \,. \end{aligned} \tag{7.12}$$

There are 16 Kraus operators in the case of two qubits. They are given by all tensor product combinations $P_1 \otimes P_2$ of the Pauli matrices $P_i \in \mathcal{P} = \{I, X, Y, Z\}$ with corresponding coefficients. The Kraus operator $I \otimes I$ is the only one with coefficient $(1 - 15/16 \cdot \lambda)$, while all others have coefficient $\lambda/15$. We do not write all 16 operators explicitly.

In our noise model, all gates are prone to depolarising error. We write $\lambda_g(q)$ for the depolarisation probability of gate g on qubit q. This yields one model parameter for each of the three single qubit gate per qubit, i.e., $3N$ parameters. Since we consider a linear qubit layout with CX gates connecting two neighboured qubits in only one direction, there are $N-1$ more parameters corresponding to the CX gate. Therefore, the total number of depolarisation parameters in our model is $4N-1$.

- **Thermal relaxation and dephasing:** We consider a qubit subject to two central dynamics as it interacts with its environment: thermal relaxation and dephasing. Both processes change the state of the qubit over time and are parametrised by the two decoherence times called T_1 and T_2, respectively. They are constrained by the condition $T_2 \leq 2 \cdot T_1$ [Paladino et al., 2011]. Since the primary cause of quantum computing time is typically gate operations, thermal relaxation and dephasing fall under the gate error category in our noise model. In the following, we distinguish between the two cases $T_2 \leq T_1$ and $T_1 < T_2 \leq 2 \cdot T_1$. The first case is discussed in full detail, while the second one requires a different approach than the straightforward definition of Kraus operators. An extensive discussion can be found in [Georgopoulos et al., 2021a].

Thermal relaxation drives qubits towards their energetic ground state. It can mathematically be described by the operators $|0\rangle\langle 0|$ and $|0\rangle\langle 1|$. The probability of this process occurring over a time interval t is $p = 1 - \exp(-t/T_1)$. Therefore, if T_g denotes the time it takes to execute a gate operation g on qubit q, the relaxation probability is

$$p_{\text{relax}} = 1 - \exp\left(-\frac{T_g}{T_1(q)}\right) . \tag{7.13}$$

Dephasing describes the destruction of relative phases between computational basis states in an overall quantum state $|\psi\rangle$, causing the loss of valuable information. When a qubit q experiences dephasing, its state is changed by the Pauli operator Z with probability

$$p_Z = \frac{\exp\left(-\frac{T_g}{T_1(q)}\right) - \exp\left(-\frac{T_g}{T_2(q)}\right)}{2} . \tag{7.14}$$

The operations and probabilities above yield the following three Kraus operators for the thermal relaxation and dephasing channel $\mathcal{T}$, where $p_I = 1 - p_{\text{relax}} - p_Z$:

$$\begin{aligned} K_0 &= \sqrt{p_I} \cdot I \, , \\ K_1 &= \sqrt{p_{\text{relax}}} \cdot |0\rangle\langle 0| \ \ , \\ K_2 &= \sqrt{p_{\text{relax}}} \cdot |0\rangle\langle 1| \ \ , \\ K_3 &= \sqrt{p_Z} \cdot Z \, . \end{aligned} \tag{7.15}$$

Thermal relaxation affects all qubits a gate g acts on in our model. The decoherence times $T_{1,2}(q)$ are part of the model parameters and can differ for each qubit q, yielding $2N$ parameters. The gate times T_g are read out from regular device calibrations.

In total, our noise model depends on $11N - 1$ parameters for linear qubit layouts, where CX gates only act in one direction. This number scales linearly with the hardware size, which is important for the practicability of parameter optimisation, see Section 7.3 below.

symbol	error	parameters	number of parameters
$\mathcal{S}$	state preparation	$p_s(q)$	N
$\mathcal{D}$	depolarization	$\lambda_g(q)$	$4N-1$
$\mathcal{C}$	crosstalk	$\phi_g(q)$	$2N$
$\mathcal{T}$	thermal relaxation	$T_1(q)$	N
		$T_2(q)$	N
$\mathcal{M}$	measurement	$p_{0\to 1}(q)$, $p_{1\to 0}(q)$	$2N$
total			$11N-1$

Table 7.1.: Parameters of full noise model.

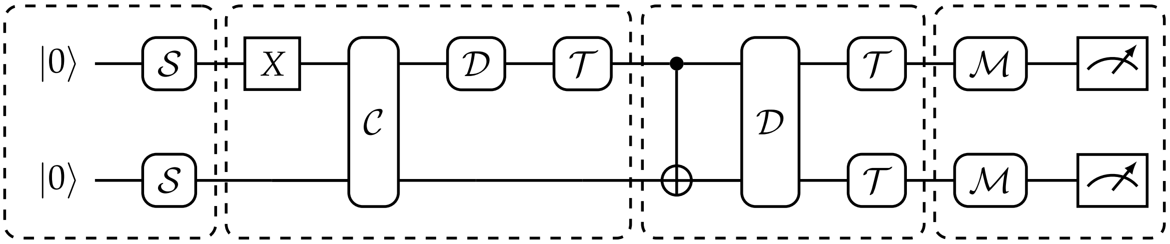


Figure 7.3.: Effect of full noise model on the example quantum circuit.

We write θ for the collection of all parameters and M_θ for the noise model with θ as its parameters.

7.2.3. Example

In the following, we briefly illustrate the effect of our noise model on the running example quantum circuit from Section 2.1.3. Figure 7.3 shows the corresponding noisy quantum circuit. The initial state preparation is followed by state preparation error, and the gate operations are affected by crosstalk, depolarisation, and thermal relaxation/dephasing. Finally, the measurements are erroneous, as discussed above. Note that the depolarising error $\mathcal{D}$ only affects a single qubit after the X gate, but both qubits after the two-qubit CX gate. Similarly, thermal relaxation and dephasing act on the same qubits.

7.3. Parameter Optimisation

This section explains the procedure for optimising the noise model parameters denoted by θ. The general idea is to use training quantum circuits and repeatedly compare model predictions to hardware data for these training circuits. After each comparison, the parameters are updated according to an optimisation algorithm.

7.3.1. Detailed procedure

The parameter optimisation consists of the following steps, which partially agree with the volumetric benchmarking routine.

1. **Training circuits:** Define a set $\tilde{C}$ of quantum circuits for training the model parameters. They are run by the quantum device D and simulated by the noise model M_θ with varying parameters configurations θ. We choose 100 `EfficientSU2` circuits with randomised rotation angles for $\tilde{C}$. They are different from the quantum circuits for the volumetric benchmarks later. Section 7.3.2 presents a more detailed discussion.

2. **Circuit transpiling:** The training circuits from step 1 are transpiled to the native gate set using the option `optimization_level=2` in Qiskit. Figure 6.6 in Section 6.3.4 shows how the layers are processed. The result of this step are 100 transpiled training circuits denoted as C.

3. **Hardware run:** The transpiled circuits $c \in C$ are run on the quantum computer to obtain the hardware results $D(c)$. Each circuit is executed $s = 8192$ times to increase the statistical significance of the samples.

4. **Set initial parameters:** The model parameters are set to an initial configuration $\theta = \theta_0$.

5. **Model predictions:** The noise model M_θ simulates the training circuits c with the current parameter configuration θ to obtain the model predictions $M_\theta(c)$. Simulations are performed using exact computations of density matrices. Implementation details can be found in Appendix B.

6. **Loss function evaluation:** The model predictions are compared to the hardware data for every training circuit using a loss function. We discuss possible choices in Section 7.3.3.

7. **Update parameters:** The model parameters are updated according to the optimisation algorithm. Section 7.3.5 presents suitable options.

Steps 4. to 7. are repeated until a satisfactory degree of convergence with respect to the loss function has been achieved.

7.3.2. Training circuits

The parameter optimisation presented above uses a set of quantum circuits for training. Since the goal is to predict noisy outcomes of quantum circuits outside the training set, it should be sufficiently large to avoid overfitting and allow for better generalisation, see [Ying, 2021]. Moreover, we expect tailored noise models for specific quantum computing applications more practical than a universal noise model able to predict quantum errors for arbitrary circuits. Optimising parameters of such a specialised noise model can be accomplished by using training circuits typical for the application of interest.

Variational quantum algorithms are the primary example applications throughout this thesis, and we aim to construct a noise model that accurately predicts quantum noise for this type of application. Therefore, we use 100 `EfficientSU2` circuits from Qiskit for parameter optimisation, where the rotation angles are again randomised. Since these quantum circuits are transpiled to the native gates as shown in Figure 6.6, the transpiled circuits do not include X gates. The only single-qubits gates after transpiling are $\sqrt{X}$ and R_z. Since the latter is implemented virtually on IBM machines, it is not erroneous, and the only single-qubit gate left in our noise model is the $\sqrt{X}$ gate. This reduces the number of parameters we optimise in practice to $8N - 1$.

7.3.3. Loss function

Step 6 of the parameter optimisation evaluates a loss function that estimates the deviation between the current model predictions and hardware data. The input os the loss function are the predictions and quantum computer results of all 100 training circuits. It needs to compute a real number that the optimisation algorithm can process for updating the parameters. Section 6.3.3 discusses suitable choices for objective functions for volumetric benchmarks, which can also be considered here because the task is very similar. Although the computation of quantum mechanical expectation values is essential in VQAs, we choose loss functions operating on a lower level and directly comparing probability distributions. We compare the outcome distributions of the hardware experiments and the noise simulation for each training circuit using the Hellinger distance (HD) f_{HD} or the KL divergence f_{KL}. We briefly recall their definitions given two discrete distributions P, Q:

$$f_{\mathrm{KL}}(P \parallel Q) = -\sum_i p_i \log\left(\frac{q_i}{p_i}\right) , \tag{7.16}$$

$$f_{\mathrm{HD}}(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_i (\sqrt{p_i} - \sqrt{q_i})^2} . \tag{7.17}$$

As discussed in the evaluation of our noise model, both choices generally lead to similar performance, and we show the results for both of them. The computation of the overall loss depends on the optimisation algorithm and its specific implementation, which is explained below. Note that the KL divergence is not defined if p_i or q_i are equal to zero, and regularisation measures are necessary for useful results. For $p_i = 0$, the contribution

error	parameters	bounds
state preparation	$p_s(q)$	$[0, 1]$
depolarisation	$\lambda_g(q)$	$[0, 1]$
crosstalk	$\phi_g(q)$	$[0, \pi/4]$
thermal relaxation	$T_1(q)$	$[0, T]$
	$T_2(q)$	$[0, 2T_1]$
measurement	$p_{0\to 1}(q)$, $p_{1\to 0}(q)$	$[0, 1]$

Table 7.2.: Parameter bounds during optimisation.

to the sum us usually evaluated as zero, while $q_i = 0$ yields a very large value. We choose 10^6 in this thesis.

7.3.4. Parameter bounds

Section 7.2.2 describes the noise channels included in our noise model. Almost all its parameters are naturally subject to certain bounds. For example, every probability is constrained by the interval $[0, 1]$. Other parameters are limited by the physical processes causing the corresponding error. These bounds must be respected by the optimisation procedure to yield a realistic noise model. Besides the physical bounds of some parameters, we artificially impose further constrains on the remaining parameters to improve the optimisation by limiting the solution space. In the following, we describe in detail the bounds used for parameter optimisation.

The state preparation error $\mathcal{S}$, measurement error $\mathcal{M}$, and depolarising error $\mathcal{D}$ depend on the corresponding probabilities p_s, $p_{0\to 1}$, $p_{1\to 0}$, and λ_g. Therefore, their values must all stay between 0 and 1. Crosstalk error in our model is described by rotations of neighbouring qubits. These rotations depend on the angles ϕ_g. Although every real number $\phi_g \in \mathbb{R}$ theoretically yields a valid noise channel, the resulting noise channel is periodic in ϕ_g with a periodicity of 2π. Moreover, we assume that only relatively small rotations are induced by crosstalk error, and we set an upper bound of $\pi/4$ during parameter optimisation. The last type of noise in our model is thermal relaxation/dephasing, which is described by the decoherence times T_1 and T_2. As described in Section 7.2.2, they are constrained by $T_2 \leq 2 \cdot T_1$, see [Paladino et al., 2011] for more details. Therefore, we choose the interval $[0, 2T_1]$ as a bound for T_2. For T_1, we choose an upper bound $T = 300\mu$s. This upper bound is significantly larger than the calibrated estimate from IBM, ensuring that the correct decoherence time is included in our optimisation bounds. Table 7.2 gives an overview of all bounds described above.

7.3.5. Optimisation algorithm

An optimisation algorithm updates the model parameters after each iteration of the procedure above. We discuss suitable choices for such an algorithm in the following. Since simulating noisy quantum circuits is computationally expensive and the effort scales exponentially with the number of qubits, reducing the necessary number of loss function evaluations is essential. Moreover, no exact gradients of the loss function with respect to the model parameters are available.

While we simulate noisy quantum circuits exactly using density matrices, stochastical approaches exist as an alternative. In this case, the optimisation algorithm must also be robust against statistical fluctuations. We present two algorithms and use them to train the parameters of our noise model. Both algorithms allow for bounds on the parameters.

SPSA

The SPSA algorithm was introduced by Spall, 1992. It describes a stochastic optimisation procedure typically used when evaluating the objective function is prone to noisy fluctuations. A prominent example in the context of quantum computing is the parameter optimisation in VQAs. Since our noisy simulations are based on density matrices, repeated runs yield the same result without fluctuations that would occur with shot-based simulations.

The algorithm approximates the gradient in a random direction at each iteration using a finite difference. It updates the parameters with respect to the probed direction based on the result. Since the finite difference is computed from two points, each iteration of the algorithm requires only two evaluations of the loss function. A detailed description of the algorithm and its hyperparameters can also be found in [Spall, 2000]. The SPSA optmisation method was used in our publication [Weber, Borras, et al., 2024] with the Hellinger distance f_{HD} as loss function.

Least-squares fit

Our second method for optimising noise model parameters is a least-squares (LS) fit to the training data. We use an algorithm from the SciPy library [Virtanen et al., 2020] based on trust regions [1], see [Conn et al., 2000]. Although the number of loss function evaluations per step varies between iterations, it can generally be compared to the number from SPSA. Different options and hyperparameters exist for this implementation of an LS optimiser. All relevant details are again described in Appendix B.

[1] https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least_squares.html

7.3.6. Training configurations

The following three configurations of optimisers and loss functions are used for training:

- **SPSA + HD:** The combination of the SPSA optimiser and Hellinger distance as loss function during training has been used in [Weber, Borras, et al., 2024], where we first presented the model construction described above.
- **LS + KL:** The other two training configurations considered in this thesis use the LS optimiser due to the practical challenges of the SPSA algorithms in this context, which we discuss later in this chapter. This specific configuration uses the KL divergence as loss function.
- **LS + HD:** This last combination for training our noise model uses the LS optimiser with Hellinger distance for evaluating the loss.

The bounds are always chosen according to Table 7.2, independent of the algorithm or loss function. Moreover, the loss functions of individual quantum circuits are combined using the ensemble evaluation in (6.6) for the least-squares optimisation.

7.4. Evaluation

This section evaluates our procedure for constructing noise models using the volumetric benchmarking framework introduced in Chapter 6. Section 7.4.1 describes the detailed benchmarking procedure and explains the Qiskit noise model we compare our trained model to. Our method to handle a potential time drift of error rates is discussed in Section 7.4.2. Afterwards, Section 7.4.3 presents the benchmark results for several combinations of optimisation algorithms and loss functions. Section 7.4.4 discusses the different optimisers from a practical standpoint regarding consistency and optimisation cost. Section 7.4.5 analyses the optimised noise model parameters, their change over time, and their impact on computations.

7.4.1. Volumetric benchmarks

After training our noise model with different configurations of optimisation algorithms and loss functions, we use volumetric benchmarks to assess the accuracy of the trained models. The benchmarking protocol used for evaluating our trained noise model is the same as in the example from Chapter 6. We briefly recall the details below.

1. **Benchmark circuits:** For each volume (w, d), we use 200 `EfficientSU2` circuits with rotation angles are drawn uniformly from the interval $[0, 2\pi)$. They do not

coincide with the 100 `EfficientSU2` circuits used for parameter optimisation, but are generated with different rotation parameters. The machine is limited to $w = 5$ qubits, and we set the same limit for the circuit depth. Therefore, we have $w, d \in \{1, 2, 3, 4, 5\}$.

2. **Circuit transpiling:** We use the transpiling option `optimization_level=2` provided in Qiskit, see Figure 6.6.

3. **Noise model predictions:** We use exact density matrix computations to obtain the noise model predictions, and the expectation value $\langle Z^{\otimes w} \rangle$ is computed as $\langle Z^{\otimes w} \rangle = \mathrm{Tr}(Z^{\otimes w} \rho)$, following (2.34).

4. **Hardware experiments:** All hardware experiments were performed using $s = 8192$ shots per quantum circuit.

5. **Single circuit evaluation:** The objective function comparing model predictions to hardware results for a single quantum circuit $c \in C(w, d)$ is the absolute difference in expectation values:

$$f(M(c), D(c)) = \left| \langle Z^{\otimes w} \rangle_{M(c)} - \langle Z^{\otimes w} \rangle_{D(c)} \right| . \tag{7.18}$$

6. **Ensemble evaluation:** The ensemble volumetric benchmark evaluation is computed as the arithmetic mean over all single-circuit results for the respective volume (w, d), see (6.4).

The Qiskit device noise model

In our evaluation, we benchmark and compare all trained noise models to a noise model specific to the IBM quantum machine provided in Qiskit. This noise model consists of a subset of the noise channels included in our noise model: All gate operations and measurements are considered erroneous, where gates are prone to depolarisation and thermal relaxation/dephasing error, and measurements at the end of a quantum circuit are affected by measurement error. Table 7.3 provides a clear comparison between the Qiskit model and ours.

7.4.2. Experiment times

Since both w and d range between 1 and 5 and we consider all possible combinations, there are 25 volumes to be tested in our benchmarks. Due to the queueing system for IBM's quantum devices, we could not run all 25 experiments simultaneously or in a sufficiently small time window to avoid a drift of noise parameters between different experiments.

	Qiskit	Full model
state preparation	X	✓
depolarisation	✓	✓
crosstalk	X	✓
thermal relaxation	✓	✓
measurement	✓	✓

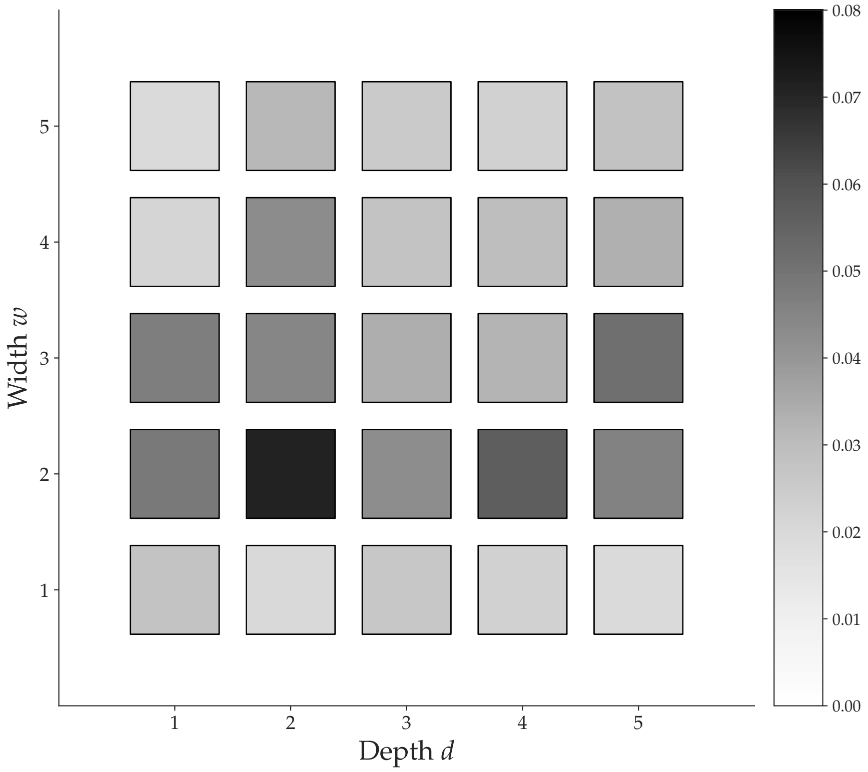
Table 7.3.: Comparison of noise channels included in different models.

Therefore, one cannot expect a single noise model to accurately describe the errors of all experiments conducted in this thesis, and model parameters must be trained or calibrated repeatedly. We address this issue by running 300 quantum circuits for each volume. Two hundred circuits are used for the volumetric benchmark, while the others are used for parameter optimisation. We use the correct corresponding Qiskit device noise model for each experiment time to enable a fair comparison. Table B.2 in Appendix B shows the experiment times.

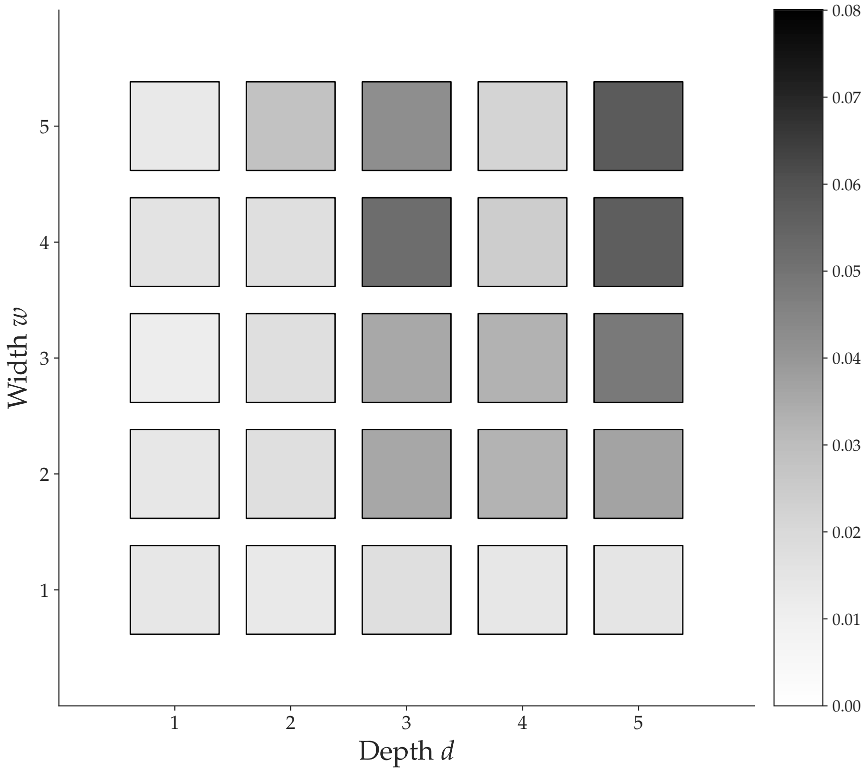
7.4.3. Benchmark results

In the following, we present the volumetric benchmarking results of our trained noise models and the one provided in Qiskit. These volumetric benchmarks form the main result of the thesis because they strongly indicate that our model construction method can successfully optimise model parameters to obtain an accurate description of the errors on a quantum computer.

Figure 7.4 shows the volumetric benchmark plots of all four noise models. Each square represents a single volume that has been tested. The colour of a square indicates the average absolute error of the Pauli expectations values described above, where darker squares correspond to larger errors and, therefore, worse model predictions. The plot clearly shows that the Qiskit device noise model performs inferior to the other models for most of the volumes. At $w = 2$ and $d = 4$, its error is dreadful. One possible cause for this phenomenon could be a malicious calibration at the corresponding time, leading to inappropriate noise parameters. However, the model from [Weber, Borras, et al., 2024] trained with SPSA and Hellinger distance also has its weaknesses for deeper circuits with more qubits. The main reason for this behaviour are difficulties arising during the training procedure for these configurations because the hyperparameter tuning of the SPSA is not straightforward. Such practical considerations are discussed later. For now, we continue analysing the benchmark results, where the SPSA optimiser achieved comparable results for many of the tested volumes. Both KL divergence and Hellinger distance have proven

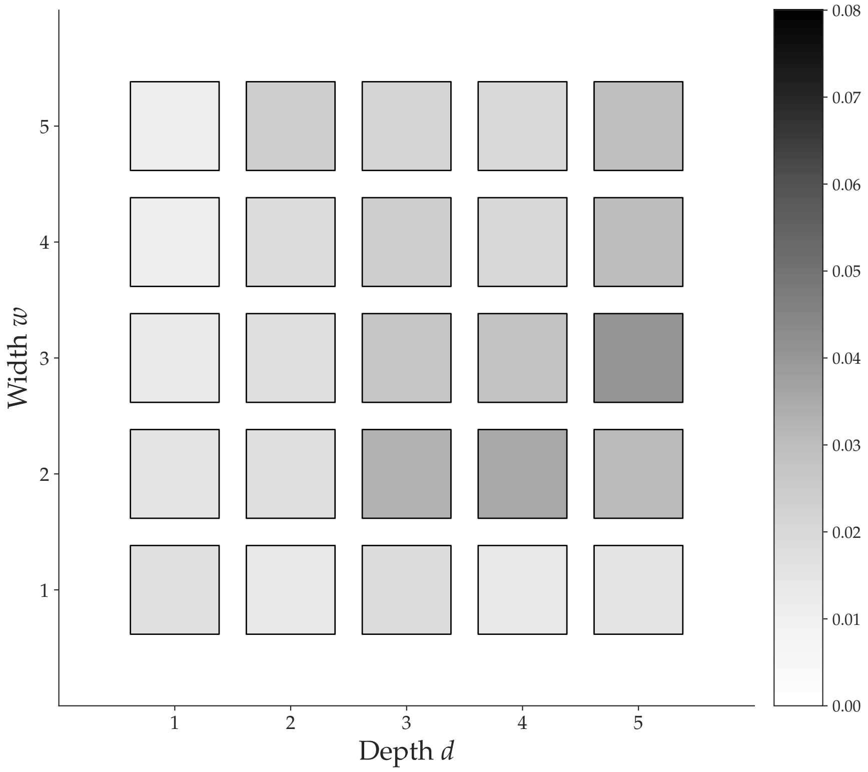


(a) `ibmq_manila` device noise model.

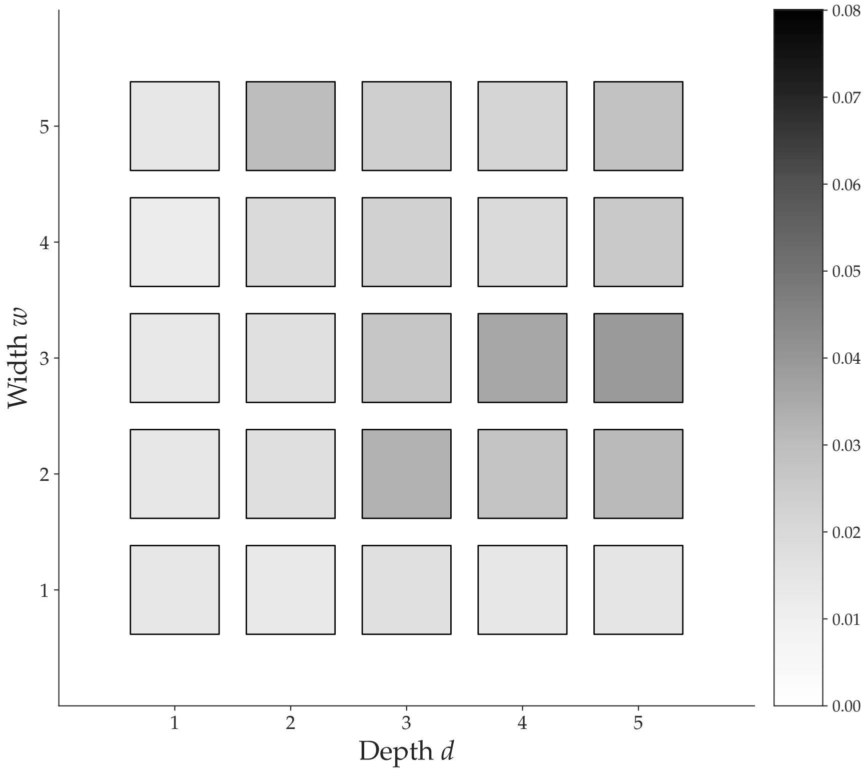


(b) SPSA + HD

Figure 7.4.: Volumetric benchmark results of the four noise models.



(c) LS+HD



(d) LS + KL

Figure 7.4.: Volumetric benchmark results of the four noise models.

to be adequate loss functions for our task.

While the volumetric plots discussed above provide a transparent qualitative overview of the results, other approaches allow a more accurate comparison of the different noise models and their benchmarks. Figure 7.5 shows the same results in a different manner, where the plot is split into five diagrams. Each diagram represents one width w and shows the average absolute error of all four noise models simultaneously for the different choices of circuit depth d.

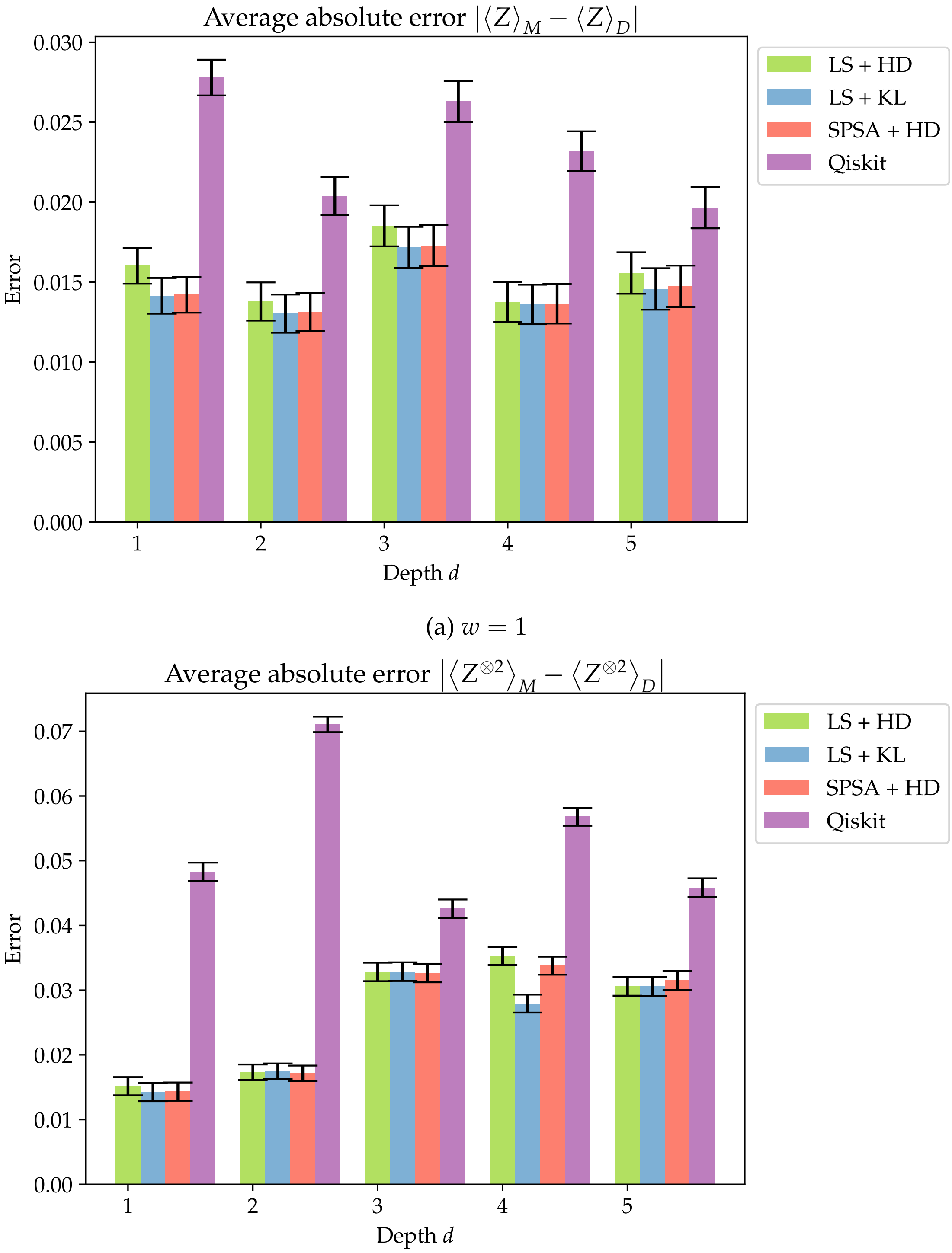
Confidence intervals: We perform bootstrapping and compute confidence intervals based on the resulting bootstrap distributions for estimating the statistical significance of the benchmark results. This paragraph explains the detailed procedure. Since we use exact density matrix computations to simulate our noise models, the only uncertainty for our benchmarks originates from the hardware results $\langle Z^{\otimes w} \rangle_{D(c)}$ of each circuit $c \in C(w, d)$. Each of these results consist of a finite samples of 8192 shots. We resample 8192 shots for every circuit by drawing *with replacement* from the original data, i.e., every point in the new sample can be any of the 8192 outcomes from the hardware. We repeat this procedure 10^5 times to generate the bootstrap distribution. The uncertainty u_c for a single quantum circuit c is estimated as the double standard deviation $u_c = 2\sigma_c$ of this bootstrap distribution. The overall error is calculated using propagation of uncertainty:

$$u = \frac{1}{200} \sqrt{\sum_c u_c^2} \,. \tag{7.19}$$

Figure 7.5 supports our findings above. The trained noise models (green with stripes, blue, and red with points) perform significantly better than the Qiskit model (purple with crosses) for almost every volume. Within the trained models, there is no overall advantage of one loss function or optimiser over the other. While the models using the Hellinger distance during training are essentially balanced in most cases, the one using KL divergence shows more variance in its performance. For example, it is significantly more accurate for $(w, d) = (2, 4)$ but less accurate for $(w, d) = (3, 5)$ or $(5, 2)$. The cause for this inaccuracy could be the regularisation needed for the case where one of the probabilities in the KL divergence is zero. The decreased accuracy of the model trained with SPSA indicated by Figure 7.4 above lays within statistical uncertainty of the other models, so that the disadvantages of the SPSA optimiser affect only the practicality during training, which is considered below.

7.4.4. Practical optmisation aspects

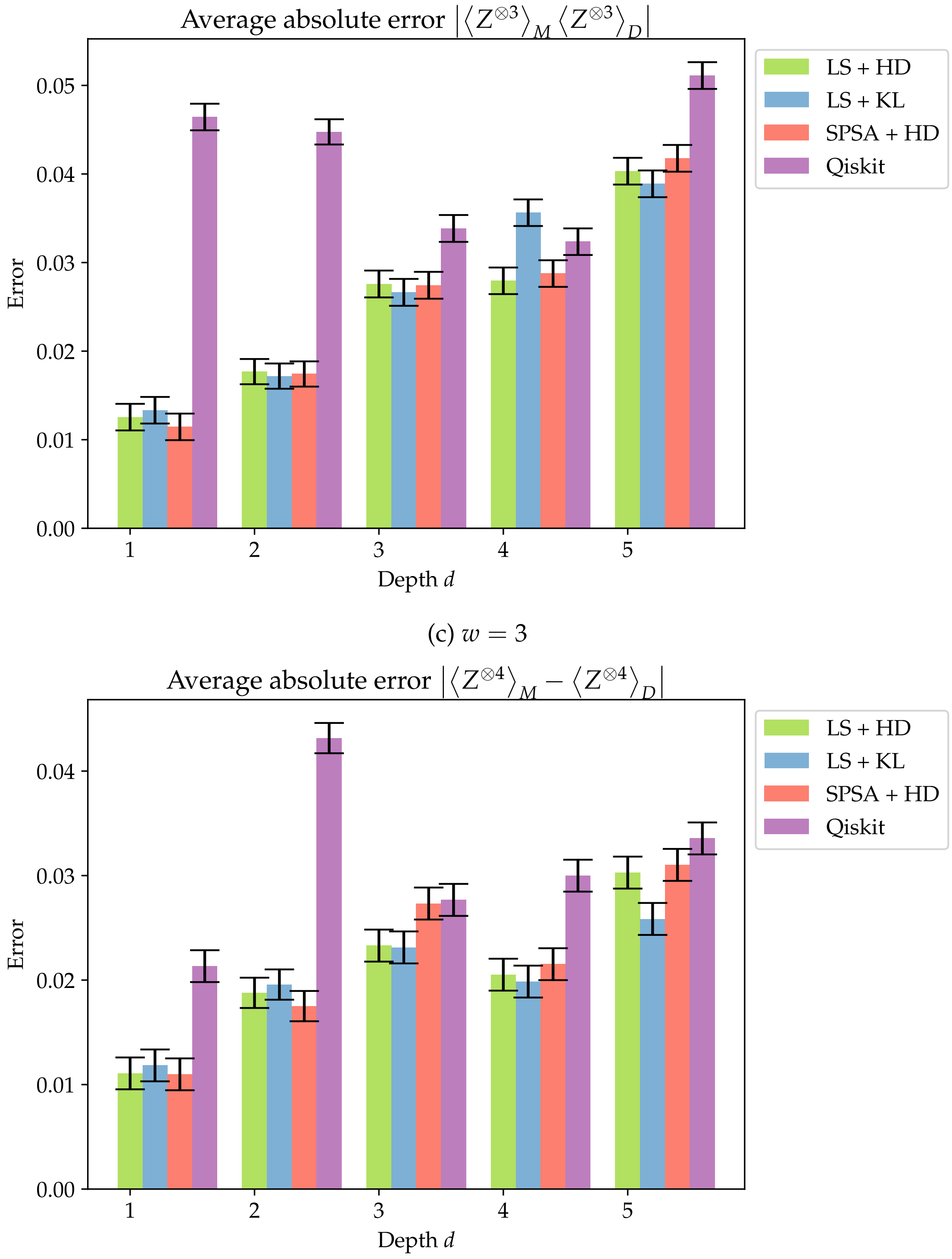
As mentioned above, we now discuss how practical the different optimisation approaches are in our setting. The main attributes we consider are the tuning of hyperparameters and



(a) $w = 1$

(b) $w = 2$

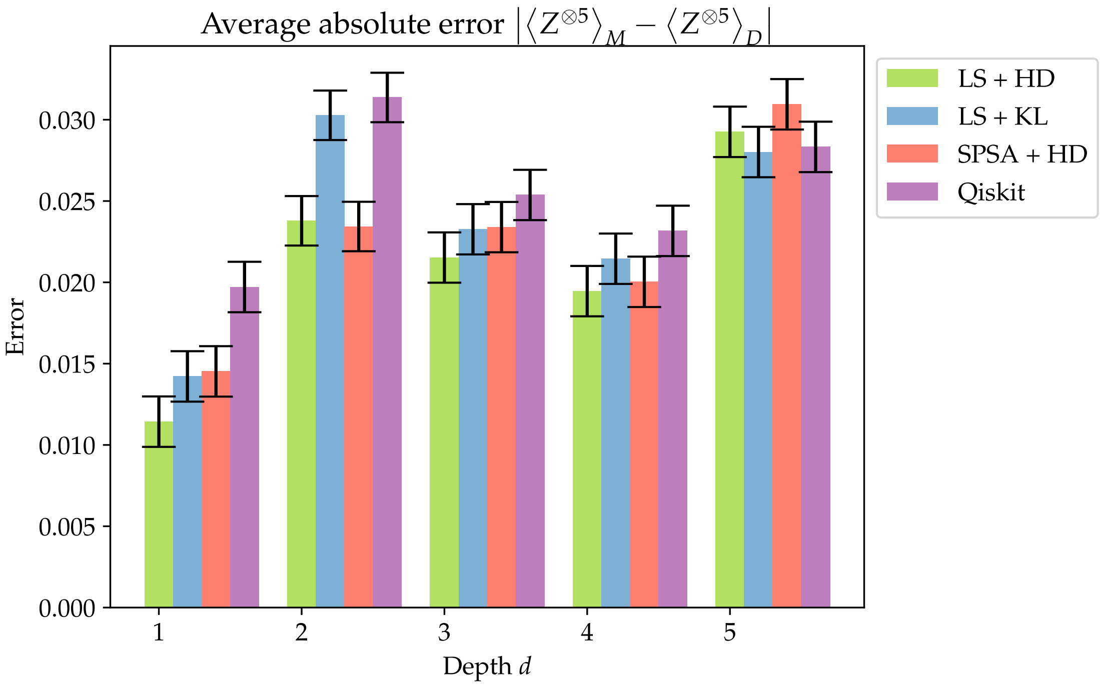
Figure 7.5.: Volumetric benchmark results for the four noise models with error bars.



(c) $w = 3$

(d) $w = 4$

Figure 7.5.: Volumetric benchmark results for the four noise models with error bars.



(e) $w = 5$

Figure 7.5.: Volumetric benchmark results for the four noise models with error bars.

the number of loss function evaluations as part of the convergence behaviour.

Optimisation algorithms: The SPSA optimiser depends on several hyperparameters, which determine the step width at each iteration and the distance of the points used for computing the gradient. Although Spall, 2000 presents a heuristic for choosing hyperparameters for the algorithm, we could not find a systematic way to achieve good convergence in our setting. This is particularly hard for increasing qubit numbers as more model parameters must be optimised. For $w = 5$, we have used 500 iterations with two loss function evaluations per iteration, yielding 1000 evaluations, which is very costly considering that larger systems are generally more interesting.

In contrast, the least-squares optimiser has produced consistently good results using significantly fewer loss function evaluations with a maximum of 50 for the largest volumes while often converging faster. Moreover, the tuning of its hyperparameters is much simpler. Although the accuracy of the resulting noise models is only slightly higher than that of the one trained with SPSA, the practicability improves significantly using this optimiser.

Loss functions: We have observed no significant difference between the two loss functions regarding their practicability during training.

7.4.5. Parameter analysis

In the following, we analyse the model parameters of the trained noise model. Since the model trained using the Hellinger distance and the least-squares optimiser has performed best in the volumetric benchmarks, we focus on this model in our analysis. The guiding questions we investigate are:

1. How do the parameters change over time?
2. Which noise channels have the largest impact on computations?

How do the parameters change over time?

We have performed the training procedure 25 times, once for each run of the quantum circuits on the machine. Therefore, one trained noise model with its optimised set of parameters exists for each of the 25 run times of the experiments. We compare these parameters for each type of error to investigate their potential drift over time. Since not every experiment involves every qubit and not every noise model contains all parameters, we restrict ourselves to the first qubit for all single-qubit errors and the CX gate between the first two qubits for all two-qubit errors. These qubits are part of the maximal number of experiments, and their errors can be investigated best. All single-qubit gate errors are understood to affect the $\sqrt{X}$ gate. Hence, the parameters considered below are:

- the bit-flip probabilities $p_{0\to 1}$ and $p_{1\to 0}$ of the readout error on the first qubit,
- the probability p_{sp} of state preparation error on the first qubit,
- the decoherence times T_1 and T_2 of the first qubit,
- the rotation angle ϕ of crosstalk error on the first qubit (affecting its neighbours, i.e., the second qubit), and
- the depolarising probabilitiy λ of the $\sqrt{X}$ gate on the first qubit, as well as the one corresponding to the CX gate from the first to the second qubit.

Note that the value of a particular parameter is simply the result of our optimisation process and does not necessarily need to correspond to an actual physical variable.

Figure 7.6 shows the time dependence of all these parameters throughout our experiments in 2022, and we discuss them separately in the following.

Readout error: The optimised bit-flip probabilities $p_{0\to 1}$ and $p_{1\to 0}$ shown in Figure 7.6a generally remain consistent over time, and the probability $p_{1\to 0}$ is usually higher than the probability $p_{0\to 1}$. However, we find two exceptions on our model parameters where the situation is reversed. In particular, the probability $p_{0\to 1}$ is noticeably high for $(w,d) = (5,2)$. The device calibration from IBM shows a different behaviour and only predicts a bit-flip probability of 1.5% compared to the value of 5.5% in our model. This discrepancy underlines that the optimal noise model parameters do not necessarily have the true physical values. Besides the outliers, the trained bit-flip probabilities agree well with the calibration data with $p_{0\to 1} \approx 1\%$ and $p_{1\to 0} \approx 3\%$ on average.

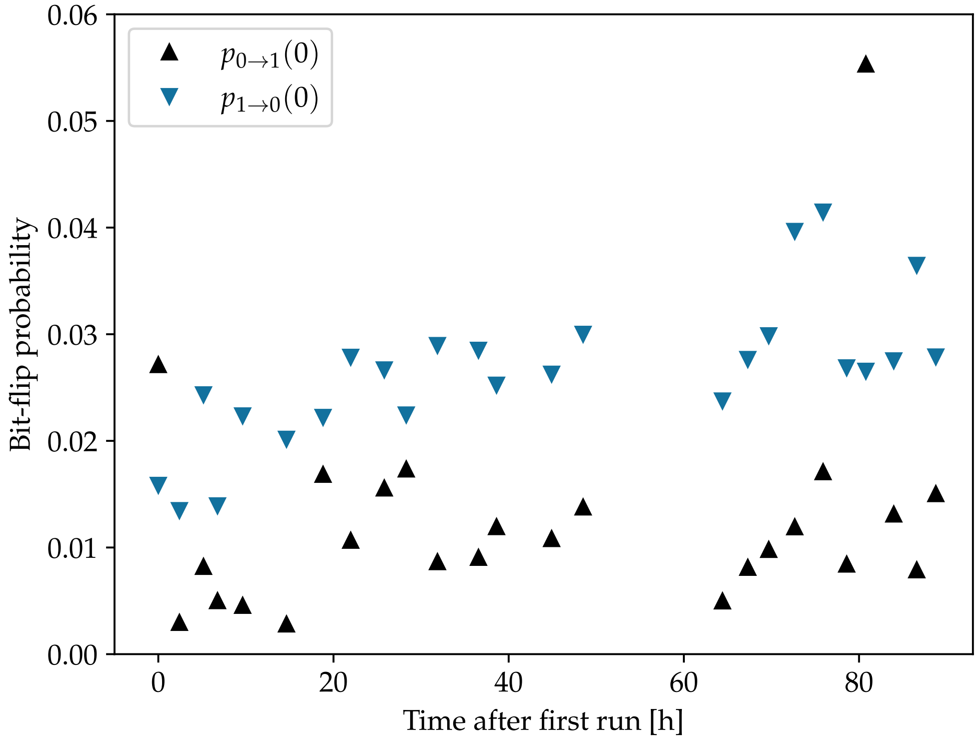
State preparation error: All trained probabilities of state preparation error are less than 1%, see Figure 7.6b. Since IBM does not account for state preparation errors in their calibration, there is no data for comparison. While the state preparation error rates mostly appear random, there is one clear peak with a maximal value of almost 1.2% after around 30 hours.

Thermal relaxation and dephasing: Except for two times, the T_1 time in the trained noise model is always larger than T_2, as presented in Figure 7.6c. While they are generally distributed around 100μs and 50μs seconds, respectively, a peak similar to one for the state preparation error states is visible 30 hours after the first experiment. However, an increase of the decoherence times means that the machine is *less* noisy, while all other noise channels become stronger with increasing parameters. The decoherence times provided by IBM are $T_1 = 131.5\mu$s and $T_2 = 102.2\mu$s for the qubit considered here. They are both slightly higher than the model parameters.

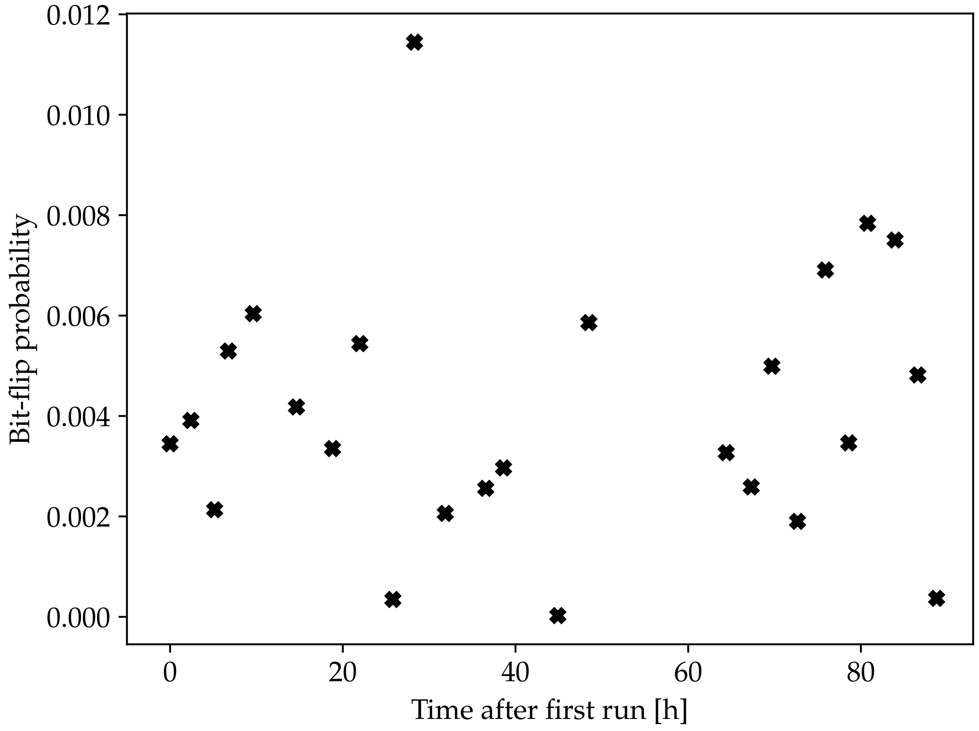
Crosstalk error: As Figure 7.6d shows, all rotation angles for crosstalk error are very small, with the largest obtained angle being $\phi \approx \pi/93$. It occurs around 30 hours after the first experiment, similar to other noise types above. As discussed later in this section, crosstalk is the least significant type of noise in the model, and it is almost negligible within the model limitations. The Qiskit noise model does not include crosstalk, and no calibration data for this type of error is available for the ibmq_manila device.

Depolarising error: Finally, the depolarising probabilities for the single-qubit $\sqrt{X}$ gate on the first qubit and the CX gate between the first two qubits are discussed, see Figures 7.6e and 7.6f. All optimised singe-qubit probabilities lie between 0% and 0.8%, while the maximal two-qubit probability in the model is 1%. This agrees very well with the calibrated error rates for the machine, which are generally also just below 1%. However, these rates do not only include depolarising error but all gate errors, making a helpful comparison to the model parameters impossible. Moreover, no pattern could be identified for the time-dependent behaviour of depolarising error. The peak after 30 hours that is present for other noise types cannot be recognised here.

Summary: The optimised noise model parameters generally show no specific pattern regarding their drift over time, except for one particular peak in the error rates for state

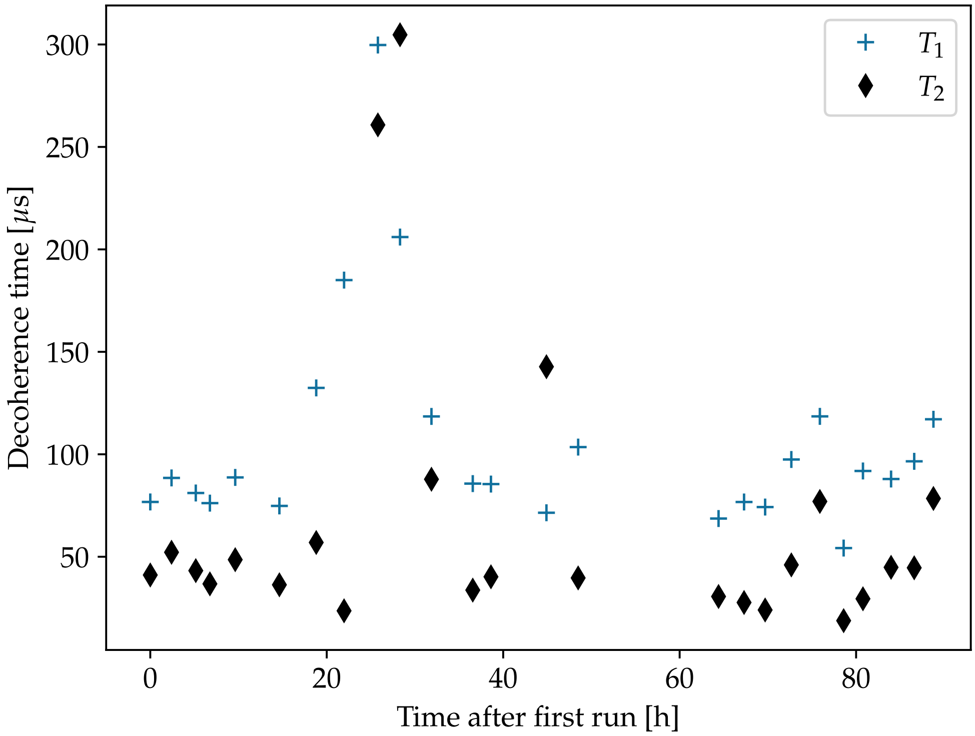


(a) Time drift of readout error probabilities $p_{0\to1}$ and $p_{1\to0}$ on the first qubit.

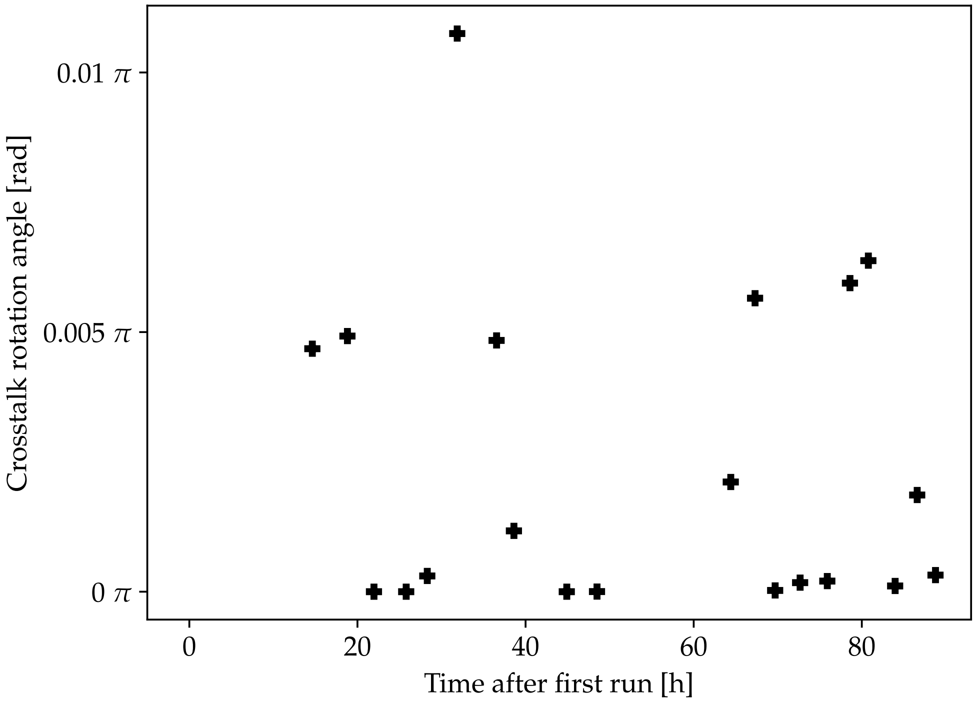


(b) Time drift of state preparation error probability p_{sp} on the first qubit.

Figure 7.6.: Noise parameters depending on experiment time.

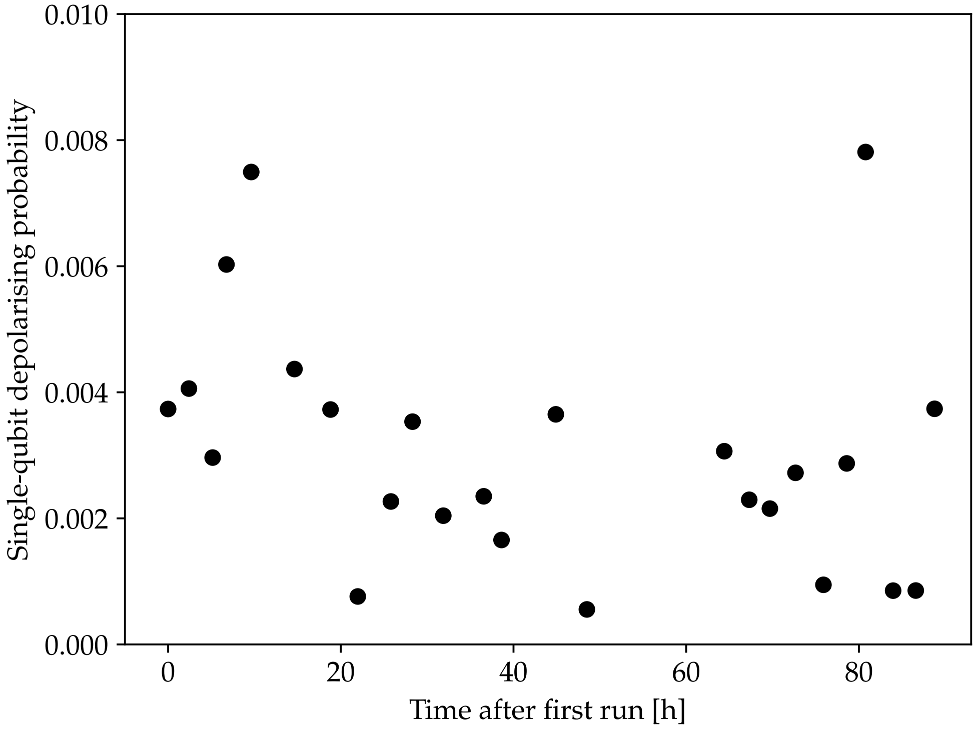


(c) Time drift of decoherence times T_1 and T_2 of the first qubit.

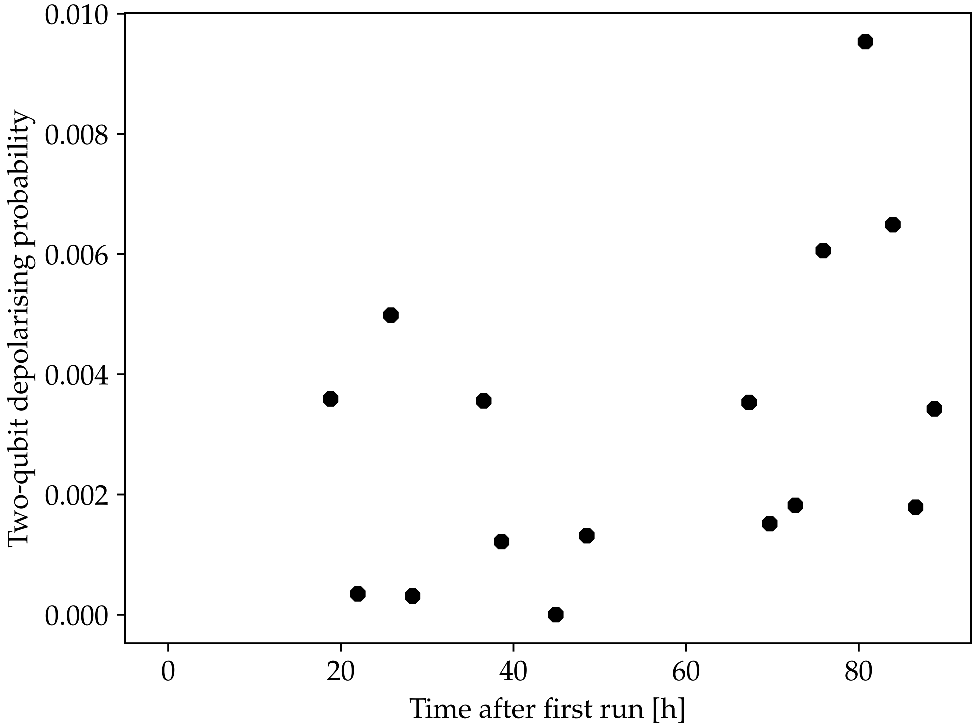


(d) Time drift of crosstalk rotation angle ϕ of the first qubit.

Figure 7.6.: Noise parameters depending on experiment time.



(e) Time drift of single-qubit depolarising error probability $\lambda_{\sqrt{X}}$ of the $\sqrt{X}$ gate on the first qubit.



(f) Time drift of two-qubit depolarising error probability λ_{CX} of the CX gate between the first two qubits.

Figure 7.6.: Noise parameters depending on experiment time.

preparation, thermal relaxation and dephasing, and crosstalk error. This peak occurs 30 hours after the first experiment and corresponds to the 10th run from Table B.2. It affects three types of noise in the trained model, namely crosstalk error, state preparation error, and thermal relaxation and dephasing. However, the noise is amplified only for the former two types of noise, and lowered for thermal relaxation and dephasing because larger decoherence times imply less errors.

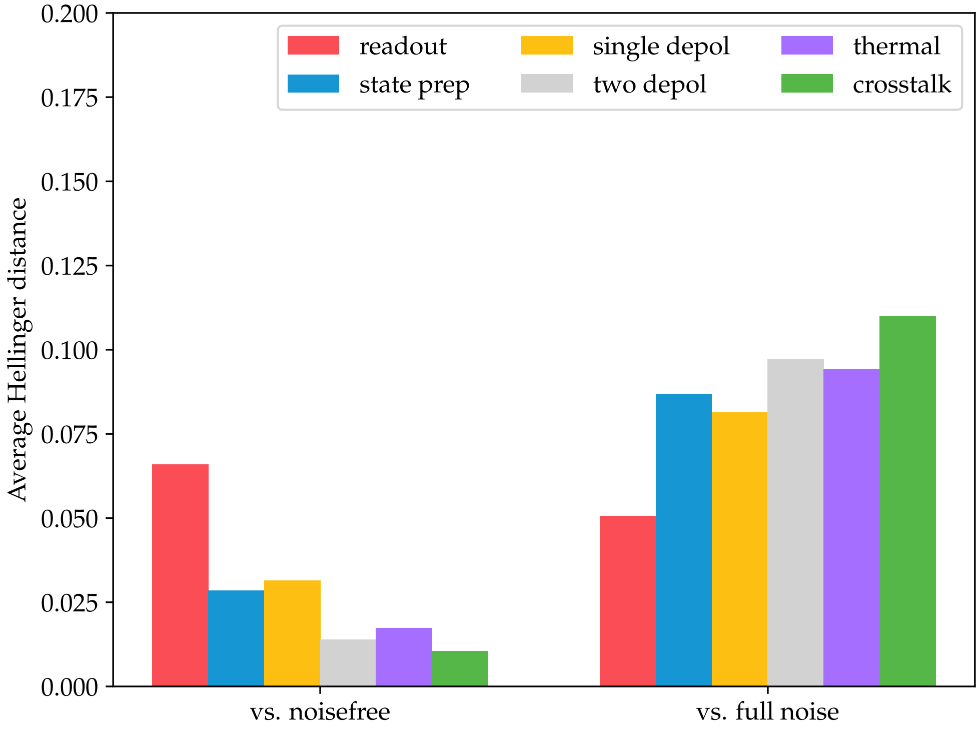
In the cases where IBM provides calibration data for certain errors, the trained parameters agree roughly with these data, at least regarding the order of magnitude. The specific values, however, deviate from the calibration.

Which noise channels have the largest impact on computations?

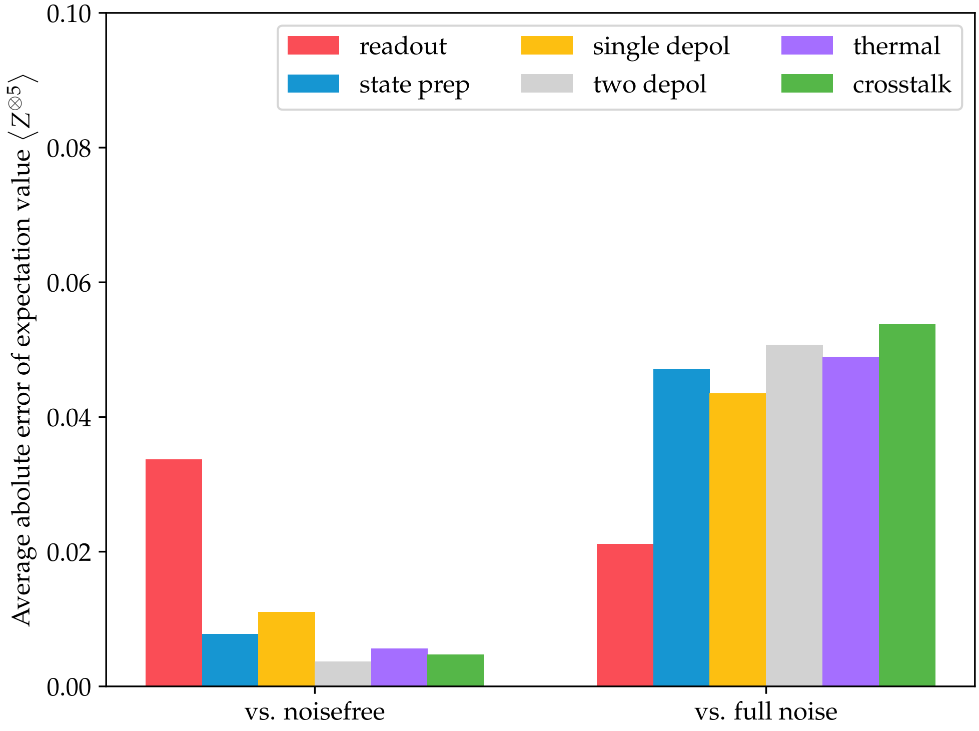
While the model parameters for all included noise types are analysed above, these parameters do not directly give insight into the impact of error on computation. If a specific type of error is very likely, it does not automatically cause a severe degradation of computation results. Therefore, this section provides a separate analysis regarding the effects of different noise types and their significance in the model.

We present the impact analysis for the configuration $(w, d) = (5, 5)$. Other volumes were also considered, but the results show almost no deviation from the ones shown in this chapter and are omitted for clarity. A partial noise model is constructed for each type of noise, including only the noise channels of interest. For example, in the case of readout error, it only consists of readout noise at the end of every quantum circuit. The impact of each error type is estimated by comparing the outcome probabilities of a simulation using the partial noise model (which only consists of one error type) to those coming from a noiseless simulation as well as the results of the full noise model that includes all noise channels. Two quantities from above are used for this comparison: the average Hellinger distance and the average absolute difference between expectation values $\langle Z^{\otimes 5} \rangle$. While the Hellinger distance directly compares the outcome distributions, the difference between expectation values indicates the degradation of computational results in a context where this quantity is of interest. Impactful noise types are expected to yield larger differences when comparing their partial noise models to the ideal case and smaller ones when compared to the full noise model. The test circuits are the 200 `EfficientSU2` circuits already used in the volumetric benchmarks earlier in this chapter.

Figure 7.7 shows the results of the impact analysis described above. The left-hand side of each plot represents the comparison between the partial noise model and the ideal case, while the right-hand side indicates the deviation between the partial noise model and the full model. The Hellinger distance is used in Figure 7.7a and the difference between expectation values in Figure 7.7b. Note that all results were obtained using exact density



(a) Average Hellinger distances between outcome distributions of partial noise model and (left) ideal case, (right) full noise model.



(b) Average absolute difference of expectation values $\langle Z^{\otimes 5} \rangle$ between outcome of partial noise model and (left) ideal case, (right) full noise model.

Figure 7.7.: Comparison of partial noise model to the ideal case and the full noise model.

matrix simulations; therefore, the plots show no error bars.

Both metrics clearly show that readout error has a significantly more detrimental impact in the trained noise model than all other types of noise. It leads to the largest deviation of computational results from the ideal ones and forms the most important source of error. It is followed in its impact by state preparation error and single-qubit depolarisation. Crosstalk and two-qubit depolarising error have the least influence on measurement outcomes on the trained noise model. The primary reason for the latter is the relatively small number of two-qubit gates in the test circuits, which are reasonably shallow. Recall from Figure 6.6 that each rotational layer in the transpiled circuits contains two $\sqrt{X}$ gates per qubit compared to the single CX gate between neighboured qubits in the entanglement layers. For deeper circuits, low two-qubit fidelities are typically a main obstacle for reliable quantum computations. This discrepancy highlights that the trained noise model effectively describes the noise device, but the noise channels do not necessarily coincide with the actual physical processes.

7.4.6. Discussion

In the remainder of this section, the validity of the methods is examined. The three main aspects concerned below are the optimisation procedure, the evaluation using volumetric benchmarks, and the analysis of model parameters with respect to their time drift and impact on quantum circuits.

Model construction: The noise model presented in this chapter considers all circuit operations erroneous and is already reasonably expressive, as proved by the volumetric benchmarks. Depending on the qubit connectivity, the number of model parameters scales linearly (as in this thesis) or maximally quadratically (for full connectivity) in the number of qubits, assuming that different error rates are possible for each two-qubit gate. This scaling is crucial for the practicability of our method because the parameters must be optimised. Common optimisation algorithms such as SPSA or least-squares are suitable for this task and the range of parameter numbers in this context.

Training the noise model requires the ability to simulate the training circuits using the model. While this is possible for small qubit numbers, challenges arise when the system size increases and noisy simulations become extraordinarily costly or impossible. To address this problem, we propose to train the noise model on varying subsets of qubits and obtain an overall noise model by merging the individual results. For example, one could use training circuits involving feasible qubit numbers and run them on overlapping qubit batches to obtain a complete device description. Such a training procedure assumes that errors are local to some extent, meaning that error rates within a batch of qubits are independent of those from another batch. This chapter has shown that the parameters of a noise model for five qubits can easily be optimised in practice, and we expect that to be possible for at least 10-15 qubits with highly efficient simulators. This would enable a feasible generalisation of our method for NISQ devices, which typically exhibit restricted connectivity and comply with the locality assumption described above. However, further experiments are necessary to prove the practicability for larger systems in practice.

Volumetric benchmarks: The volumetric benchmarking framework for noise models in general is discussed comprehensively in Chapter 6. Here, we focus on the specific choices for loss function and test circuits to estimate the validity of our evaluation.

The quantum circuits used for benchmarking are `EfficientSU2` circuits with alternating layers of rotation and entanglement gates. Hence, they are highly structured and errors could propagate through these circuits differently than for more generic ones. Since the parameter were optimised using the same type of quantum circuit (but with different rotations), the trained noise models could potentially be favoured in the benchmarks. On the other hand, the circuits incorporate all relevant quantum gates of the machine equally, so gate errors accumulate over the circuit to the same extent for single and two-qubit gates. Furthermore, the Qiskit noise model is calibrated with additional hardware parameters that are unaccessible during our training procedure; for example, the decoherence times T_1 and T_2 can be obtained more directly. More experiments with different quantum circuits could yield a better understanding of the generalisability of our results, but the quantum computing resources for this work were limited.

In order to measure the deviation between model predictions and hardware results, we used the absolute difference between expectation values of the $\langle Z^{\otimes w} \rangle$ observable. This measure does not directly give insight into the accuracy of the noise models for other types of computations but only detects how well expectation values are predicted. However, many quantum computing applications designed for near-future devices rely on these expectation values, making the measure a helpful quantity. Additional benchmarks show that the model predictions also agree well with hardware behaviour when using the Hellinger distance or KL divergence. Since the models are trained with these objective functions, using them for our evaluation would have led to an unfair comparison to the Qiskit noise model. Moreover, the results show that parameter optimisation can be achieved with general loss functions (HD and KL divergence) unrelated to any later use cases of the noise model.

Parameter and impact analysis: Analysing the optimised model parameters is only helpful to a limited extent. Firstly, although the noise channels in the model correspond to processes that are likely to occur on the hardware, the trained parameters do not necessarily correspond to the rates of these processes but only yield the best possible effective description within the limits of the noise model. However, they surprisingly agree with calibration data (they are estimated to the same order of magnitude by the model, which is not obvious) whenever such comparisons are possible. Secondly, the time drift of model parameters appears to be predominantly random, except for a single peak of specific error rates. State preparation and crosstalk error increase at this peak while the thermal relaxation and dephasing are suppressed (higher decoherence times at the peak mean less noise). The most likely reason is that the trained model reorganises the total noise differently into these channels. The resulting noise does not necessarily need to yield different outcomes for the quantum circuits because of their layered structure.

Investigating the impact of different noise channels for the trained models shows that readout error has, by far, the most detrimental effect on computations. This holds primarily for the quantum circuits used in this section but coincides with observations in the literature, see [Funcke, Hartung, Jansen, Kühn, Stornati, and Wang, 2022] and [Bravyi et al., 2021]. Since these circuits are all relatively shallow, readout error is also expected to play a significant role. Gate errors accumulate much more in deeper circuits. Although there are different error mitigation schemes counteracting gate errors [Li and Benjamin, 2017; Temme et al., 2017], they all suffer from an exponential computational overhead with respect to the number of gate layers. Takagi et al., 2022 have shown that this phenomenon is inevitable for all error mitigation methods because the variance of the results increases after mitigation increases. Such a variance increase can also be observed in practice for measurement error mitigation, as presented in our work in [Alexandrou et al., 2022]. However, the readout noise does not depend on the circuit depth, and since readout error in the present scenario is the most dominant type of noise, error mitigation can still be helpful.

7.5. Summary

This chapter introduced a method to construct noise models for quantum computing. The two primary steps for this construction are choosing a set of noise channels and optimising their parameters. Volumetric benchmarks were then used to evaluate the accuracy of the optimised noise model. Moreover, model parameters were analysed, and the impact of different noise types on computational results was discussed.

The particular noise model we presented in this chapter consists of state preparation error, thermal relaxation and dephasing, depolarising error on single and two-qubit gates, and readout error at the end of every quantum circuit. We optimised model parameters by repeatedly simulating training circuits, comparing the resulting model predictions to data from an actual quantum device, and updating the model parameters accordingly. A set of 100 `EfficientSU2` circuits with randomised rotation angles was used for training. The training procedure was repeated 25 times, corresponding to the 25 experiment times on the quantum hardware.

Quantum circuits with different rotation angles acted as test circuits for a volumetric benchmarking of the trained noise models. The benchmark results show a significantly higher accuracy of the optimised model compared to the Qiskit one and minor improvements over the model presented in [Weber, Borras, et al., 2024]. Therefore, this chapter presented a proof of principle of the training method using least-square optimisation and layered quantum circuits.

Analysing the drift of model parameters over different experiment times showed predominantly random behaviour. The only remarkable increase in error rates occurred approximately 30 hours after the first experiment for state preparation and crosstalk error. However, decoherence times in the model increased simultaneously, meaning that the overall noisiness of the device can hardly be estimated. Generally, the optimised model parameters agreed roughly with calibration data when such data existed.

An examination of the impact of different noise types in the trained model on computational results exhibited a dominance of readout error over all other errors. In the simulations performed in this chapter, crosstalk and two-qubit depolarising errors seem almost negligible, which is certainly unusual for the latter one. However, thermal relaxation and dephasing affecting two-qubit gates and their relatively low fidelities often reported in the literature could be absorbed into this type of noise by the model.

In summary, this chapter presented the following main contributions to this thesis:

- A set of noise channels that can effectively describe noisy quantum hardware when optimised.
- An optimisation procedure for the model parameters, including suitable training circuits, objective functions, and optimisation algorithms.
- A systematic evaluation of the trained noise model using the volumetric benchmarks introduced in Chapter 6.
- An analysis of the model parameters and noisy channels regarding their drift over time and their impact on computations.

CHAPTER

EIGHT

CONCLUSION

The following chapter concludes this thesis by providing a comprehensive overview of all contributions in Section 8.1 and an outlook on potential future research directions related to the thesis in Section 8.2.

8.1. Summary

This section summarises the results of this thesis and explains how they answer the three main research questions posed in Chapter 4. They are the focus of Chapters 5, 6, and 7, respectively, and we discuss them separately below.

RQ1: How can noise models and their impact on quantum circuits be visualised?

In Chapter 5, we introduced the concept of NQCs as an extension of the ordinary quantum circuit model by general quantum channels. While the latter only includes unitary gate operations, our extension can contain arbitrary completely positive, trace-preserving maps in computations. These operations are the mathematical formalism to model quantum errors, and almost all prevalent error types can be described this way. Therefore, our approach provides a means to represent noise graphically. In contrast to existing literature, we give a formal definition of NQCs in terms of DAGs. Noisy simulations can then easily be derived as a sequence of quantum channels applied to an initial density matrix following the DAG structure.

Ordinary quantum circuits without noise are a special case of noisy quantum circuits because unitary operations are particularly also quantum channels. Therefore, the NQC approach does not demand any changes to the existing model but is an actual extension. Non-unitary noise channels are represented by slightly different boxes, and dashed lines can be used to indicate that noise channels correspond to certain circuit operations. With this in mind, one can define a noise model as a map from (ordinary) quantum circuits to noisy ones.

In summary, noisy quantum circuits provide a practical tool for modelling noise in quantum computing. They are an intuitive extension of the widely used existing circuit model, and noisy simulations can (in theory) be directly derived from noisy quantum circuits.

RQ2: How can noise models be evaluated and compared regarding their accuracy?

Chapter 6 addresses the lack of a systematic benchmarking approach for evaluating the accuracy of noise models in quantum computing. To fill this gap, it introduces volumetric benchmarks, which were initially published in the literature for measuring the reliability of quantum hardware. Volumetric benchmarks compare model predictions obtained from simulations to hardware data for test quantum circuits with varying volumes, i.e., numbers of qubits and circuit depths. An objective function measures the distance between the corresponding outcome distributions. We discuss several choices for the circuits and the objective function.

Another essential aspect of the research question above is ensuring the quality of our benchmarking approach. Since benchmarks for classical computing systems have existed for a long time, there are established quality criteria in the literature. The criteria of interest for this thesis are relevance, reproducibility, fairness, verifiability, and usability. They all need to be adapted to the present setting of noisy quantum computers, and this thesis provides adequate interpretations. Using these criteria, we evaluate the volumetric benchmarking approach and explain how the quality can be ensured with appropriate circuits and objective functions.

Volumetric benchmarks are a flexible tool for estimating the accuracy of noise models and comparing them to each other. This thesis shows that they are an adequate approach for this task and can achieve high quality regarding the criteria.

RQ3: How can realistic noise models be constructed?

In Chapter 4, the primary goal of this thesis was defined as follows:

> **Construct an understandable noise model that is more realistic than existing noise models from the literature.**

While the chapters answering the first two research questions contain crucial contributions, this final research question targets the essence of achieving this goal. It is the focus of Chapter 7, where we construct a noise model containing five types of errors and optimise the model parameters. The training procedure consists of repeatedly evaluating model predictions against hardware data for a set of training circuits and updating the model parameters accordingly. We discuss different objective functions for this evaluation, suitable quantum circuits for training, and appropriate optimisation algorithms. Afterwards, we use volumetric benchmarks to measure the accuracy of several trained noise models and compare them to each other and the Qiskit noise model for the machine. The results show that high accuracies can be achieved with our optimisation strategy. In particular, all the trained noise models describe the quantum device better than the Qiskit model.

We also analyse our most successful noise model regarding the impact of different errors on computational results and the time drift of error rates. The latter reveals mostly random behaviour of the model parameters within a range that agrees with other existing calibration data. However, at some points during the experiment times, the trained noise model exhibits increased error rates for crosstalk and state preparation error and simultaneously lower error rates for thermal relaxation and dephasing. On the other hand, the impact analysis shows that readout error is the dominant source of noise in our model, followed by single-qubit depolarisation and state preparation error.

Finally, we can assess the question of whether this thesis has accomplished the overall research goal stated above. We have presented an easily extendable noise model and a practical method for optimising its parameters. The model is understandable because it only includes noise channels representing intelligible errors such as readout errors. A systematic evaluation using volumetric benchmarks shows that it is more accurate than existing noise models from the literature. The choice of training and benchmark circuits ensures that it is instrumental in the context of measuring quantum mechanical observables on a quantum computer. Therefore, the research goal of this thesis has indeed been achieved.

8.2. Outlook and future work

The three main chapters of this thesis are concerned with visualising noise, benchmarking noise models, and constructing noise models. We discuss potential future research directions for each of these aspects in the following.

Visualising quantum noise

Several computer tools exist that offer a graphical interface for constructing quantum circuits. These tools often enable the user to convert the graphical model to a set of instructions described in a quantum assembly language such as OpenQASM [Cross, Bishop, Smolin, et al., 2017]. Such instructions can be processed by quantum computers or directly translated into classical simulation programs. Such routines could also be developed for noisy quantum circuits, making it straightforward to investigate quantum noise and perform noisy simulations. Our formal definition of noisy quantum circuits makes the approach very harmonious with that task and other potential integrations into software.

Benchmarking noise models

Since volumetric benchmarks for quantum computing noise models were just recently at the time of this thesis (see [Weber, Borras, et al., 2024]), two potential directions for future work arise.

Firstly, the benchmarks themselves could be improved. New objective functions that capture the model accuracy well for specific use cases of quantum computing should be investigated. Moreover, a collection of quantum circuits could be curated for more general benchmarks not tailored to particular applications. The goal should be to increase the quality of the benchmarks in terms of the criteria proposed in this thesis, with a special focus on relevance, which captures the generalisability of the benchmarks.

Secondly, extensive benchmarks of existing noise models should be performed to enable a fair comparison. The results could be collected in a publicly available database to ensure full transparency for researchers and industries interested in NISQ hardware. A software suite for automatised volumetric benchmarks would enormously benefit such a project.

Constructing noise models

As explained in Chapter 7, constructing a noise model consists of two steps: choosing a set of noise channels and optimising their parameters. These two steps naturally lead to two open fields for future research.

Additional noise channels could extend the noise model presented in this thesis. We use a relatively basic model of crosstalk error here, and more sophisticated types of crosstalk could be incorporated. Moreover, hardware-specific channels could be added, depending on the quantum computing architecture of interest. Non-Markovian errors could also be considered, although they require larger modifications on how we generally describe noise in this thesis.

The second main direction for future work on constructing noise models concerns parameter optimisation. Similar to the volumetric benchmarks, new loss functions and training circuits should be investigated. The latter are ideally general circuits that allow us to obtain accurate noise models for different applications of quantum computing without adapting the training circuits. However, there is a more critical challenge to tackle regarding model training. As discussed in Chapter 7, noisy simulations become exponentially costly for increasing qubit numbers. Therefore, a practical solution that evades this problem must be developed by training the model on subsets of qubits and interweaving the results for overlapping qubit batches. While this assumes some degree of locality of the errors, we expect that the optimisation method in this thesis can successfully be generalised to larger system sizes.

APPENDIX

A

DIRAC NOTATION

The *Dirac notation* (also known as bra-ket notation) was introduced by Dirac, 1939 and is widely used in quantum mechanics. This chapter explains all symbols that appear throughout this thesis. It follows Hall, 2013, where the reader can find a more extensive discussion.

Let $\mathcal{H}$ be a complex Hilbert space describing a physical system, for example, a set of qubits. For two vectors $\psi, \phi \in \mathcal{H}$, denote their scalar product by $\langle \psi, \phi \rangle$. As the name *bra-ket* suggests, brackets of different forms are used to describe vectors and scalar products. There are two types of brackets, which are named *bra* and *ket*. A ket is simply a vector $\psi \in \mathcal{H}$, and it is written as $|\psi\rangle$. On the other hand, a bra is a continous linear functional $\mathcal{H} \to \mathbb{C}$. If ϕ is a vector in $\mathcal{H}$, then $\langle \phi|$ is the functional mapping a ket $|\psi\rangle$ to the scalar product

$$\langle \phi| \left(|\psi\rangle\right) = \langle \phi, \psi \rangle = \langle \phi|\psi\rangle \ .$$

As an example, consider a single qubit with its two-dimensional Hilbert space $\mathbb{C}^2$. The canonical basis is given by the two vectors

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix} \ .$$

They are typically denoted by the kets $|0\rangle$ and $|1\rangle$, respectively. Their scalar product is zero:

$$\langle 0|1\rangle = \langle 1|0\rangle = 0 \, .$$

If $A\colon \mathcal{H} \to \mathcal{H}$ is a linear map, then we can consider the bracket between ϕ and $A\,|\psi\rangle$, which is then denoted by $\langle\phi|A|\psi\rangle$. In the case that A is a physical quantity of interest, i.e., an observable, the bracket $\langle\psi|A|\psi\rangle$ is called the *expectation value* of A in the state $|\psi\rangle$. For example, let

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

be the Pauli matrix from Chapter 2. The expectation value of this observable in the computational basis states is

$$\begin{aligned}
\langle 0|Z|0\rangle &= \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 1\,, \\
\langle 1|Z|1\rangle &= \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = -1\,,
\end{aligned}$$

which directly correspond to the matrix elements of Z.

APPENDIX

B

IMPLEMENTATION DETAILS

This section presents in more detail how we implemented different computational tasks throughout this thesis. It describes the software for running quantum circuits on the quantum hardware and simulating them with noise models. Moreover, it contains an overview of all experiment times in September 2022, when the experiments on the IBM hardware were carried out. An overview over all important software packages and their versions can be found in Table B.1.

Quantum Computations

We used Qiskit to create quantum circuits, transpile them to native gates, and run them on IBM's ibmq_manila device. All transpiler calls used the option `optimization_level = 2` to reduce the total number of native gates. We manually checked that the quantum circuits were transpiled equally for all volumes and circuits, i.e., that rotation layers were always transformed to the same native gates. Before running the circuits on the hardware, we saved all transpiled circuits using the Open Quantum Assembly Language (OpenQASM) for later noisy simulations. Afterwards, we ran the quantum circuits on the machine, where 300 circuits for each of the 25 volumes (w, d) defined in Chapter 6 were run in one job. Since there was a queueing system for the quantum device, the 25 jobs were run at different times. These times are listed in Table B.2. We used $s = 8192$ shots per quantum circuit to reduce the shot noise and decrease the variance of our results.

[1] https://docs.quantum.ibm.com/api/qiskit/0.38

[2] https://github.com/numpy/numpy/releases/tag/v1.24.3

[3] https://github.com/matplotlib/matplotlib/releases/tag/v3.7.2

[4] https://github.com/scipy/scipy/releases/tag/v1.10.1

[5] https://github.com/andim/noisyopt/releases/tag/0.2.2

Software	Version
Qiskit	0.38.0 [1]
Numpy	1.24.3 [2]
Matplotlib	3.7.2 [3]
SciPy	1.10.1 [4]
NoisyOpt	0.2.2 [5]

Table B.1.: Versions of all relevant software libraries used in this thesis.

	$d = 1$	$d = 2$	$d = 3$	$d = 4$	$d = 5$
$w = 1$	26th, 19:58	26th, 22:23	27th, 01:08	27th, 02:44	27th, 05:36
$w = 2$	27th, 10:36	27th, 14:47	27th, 17:56	27th, 21:46	28th, 00:17
$w = 3$	28th, 03:50	28th, 08:31	28th, 10:35	28th, 16:52	28th, 20:28
$w = 4$	29th, 12:29	29th, 15:16	29th, 17:39	29th, 20:38	29th, 23:52
$w = 5$	30th, 02:33	30th, 04:45	30th, 07:56	30th, 10:33	30th, 12:44

Table B.2.: Execution times (UTC+2) of hardware experiments in September 2022.

Noisy simulations

As described in Chapter 1, the state of a quantum system consisting of N qubits is generally represented by a $2^N \times 2^N$ density matrix ρ. All unitary gates and noisy processes act as CPTP maps on this density matrix. Since every CPTP map can be written as

$$\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger \quad \text{with} \quad \sum_i E_i^\dagger E_i = I$$

by Theorem 2.2.3, noisy simulations of quantum circuits can be performed as a sequence of matrix operations. We used Numpy arrays in Python 3 to implement these matrix multiplication and addition operations for all noisy simulations except the Qiskit noise model. This noise model is a Qiskit-specific class, and we simulated it using the AerSimulator with $s = 8192$ shots per circuit because Qiskit offered no option to compute the full density matrix. All loss functions, i.e., expectation values, Hellinger distances, and KL divergences, were calculated with our implementations.

Parameter optmisation

We trained our noise models in Chapter 7 using two different optimisation algorithms: SPSA and least-squares. For the SPSA optimiser, we used an implementation contained in the NoisyOpt package for Python, while the least-squares implementation is part of the

Parameter	Value
n_{iter}	500
a	0.005-0.08, depending on (w, d)
c	0.005
α	0.602
γ	0.101

Table B.3.: Hyperparameters used for parameter optimisation with SPSA

SciPy library. In the following, we describe the hyperparameters and options used for the optimisation.

SPSA: The SPSA optimiser depends on various hyperparameters, see [Spall, 1992]:

- the number n_{iter} of iterations or epochs
- a parameter a describing the step size for adapting the parameters in each iteration
- a parameter c describing the distance of the two points used to approximate the gradient
- a parameter α describing the scaling of the step size a depending on the iteration
- a parameter γ describing the scaling of the gradient distance depending on the iteration

Spall, 2000 gives heuristics on how to choose these parameters. Table B.3 shows our hyperparameter choices, where a and c were obtained by trial and error, while the values of α and γ are the recommendations from the reference above.

Least-squares: There is a plethora of options to modify the least-square optimiser provided by the SciPy library. While we restricted the maximal number of objective function evaluations to 60, the algorithm converged to a solution with fewer iterations in almost all cases. We left most options at their default values, only changing the trust-region solver to `tr_solver='lsmr'` and how to compute the overall loss from all circuits to `loss='arctan'`, which reduces the influence of outliers in the training circuits.

REFERENCES

Aharonov, Dorit and Michael Ben-Or (2008). "Fault-Tolerant Quantum Computation with Constant Error Rate". *SIAM Journal on Computing* 38.4, pp. 1207–1282. DOI: 10.1137/S0097539799359385.

Aharonov, Dorit, Alexei Kitaev, and Noam Nisan (1998). "Quantum circuits with mixed states". *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. STOC '98. New York, NY, USA: Association for Computing Machinery, pp. 20–30. DOI: 10.1145/276698.276708.

Aharonov, Y., L. Davidovich, and N. Zagury (1993). "Quantum random walks". *Physical Review A* 48.2, pp. 1687–1690. DOI: 10.1103/PhysRevA.48.1687.

Ahsan, Muhammad, Syed Abbas Zilqurnain Naqvi, and Haider Anwer (2022). "Quantum circuit engineering for correcting coherent noise". *Physical Review A* 105.2, p. 022428. DOI: 10.1103/PhysRevA.105.022428.

Alexandrou, Constantia et al. (2022). "Investigating the variance increase of readout error mitigation through classical bit-flip correction on IBM and Rigetti quantum computers". *Proceedings of The 38th International Symposium on Lattice Field Theory — PoS(LATTICE2021)*. Vol. 396. SISSA Medialab, p. 243. DOI: 10.22323/1.396.0243.

Arute, Frank et al. (2019). "Quantum supremacy using a programmable superconducting processor". *Nature* 574.7779, pp. 505–510. DOI: 10.1038/s41586-019-1666-5.

Bang-Jensen, Jørgen and Gregory Z. Gutin (2009). *Digraphs*. Springer Monographs in Mathematics. London: Springer. DOI: 10.1007/978-1-84800-998-1.

Barnes, Jeff P., Colin J. Trout, Dennis Lucarelli, and B. D. Clader (2017). "Quantum error-correction failure distributions: Comparison of coherent and stochastic error models". *Physical Review A* 95.6, p. 062338. DOI: 10.1103/PhysRevA.95.062338.

Bassi, Angelo and Dirk-André Deckert (2008). "Noise gates for decoherent quantum circuits". *Physical Review A* 77.3, p. 032323. DOI: 10.1103/PhysRevA.77.032323.

Becker, Colin Kai-Uwe, Nikolay Tcholtchev, Ilie-Daniel Gheorghe-Pop, Sebastian Bock, Raphael Seidel, and Manfred Hauswirth (2022). "Towards a Quantum Benchmark Suite with Standardized KPIs". *2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C)*. 2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C), pp. 160–163. DOI: 10.1109/ICSA-C54293.2022.00038.

Beer, Kerstin et al. (2020). "Training deep quantum neural networks". *Nature Communications* 11.1, p. 808. DOI: 10.1038/s41467-020-14454-2.

Berg, Ewout van den, Zlatko K. Minev, Abhinav Kandala, and Kristan Temme (2023). "Probabilistic error cancellation with sparse Pauli–Lindblad models on noisy quantum processors". *Nature Physics*, pp. 1–6. DOI: 10.1038/s41567-023-02042-2.

Bergholm, Ville et al. (2022). *PennyLane: Automatic differentiation of hybrid quantum-classical computations*. DOI: 10.48550/arXiv.1811.04968.

Bernstein, Ethan and Umesh Vazirani (1997). "Quantum Complexity Theory". *SIAM Journal on Computing* 26.5, pp. 1411–1473. DOI: 10.1137/S0097539796300921.

Blume-Kohout, Robin, John King Gamble, Erik Nielsen, Jonathan Mizrahi, Jonathan D. Sterk, and Peter Maunz (2013). *Robust, self-consistent, closed-form tomography of quantum logic gates on a trapped ion qubit*. DOI: 10.48550/arXiv.1310.4492.

Blume-Kohout, Robin, Kenneth Rudinger, Erik Nielsen, Timothy Proctor, and Kevin Young (2020). *Wildcard error: Quantifying unmodeled errors in quantum processors*. DOI: 10.48550/arXiv.2012.12231.

Blume-Kohout, Robin and Kevin C. Young (2020). "A volumetric framework for quantum computer benchmarks". *Quantum* 4, p. 362. DOI: 10.22331/q-2020-11-15-362.

Boixo, Sergio et al. (2018). "Characterizing quantum supremacy in near-term devices". *Nature Physics* 14.6, pp. 595–600. DOI: 10.1038/s41567-018-0124-x.

Bravyi, Sergey, Sarah Sheldon, Abhinav Kandala, David C Mckay, and Jay M Gambetta (2021). "Mitigating measurement errors in multiqubit experiments". *Physical Review A* 103.4, p. 42605. DOI: 10.1103/PhysRevA.103.042605.

Breuer, Heinz-Peter and Francesco Petruccione (2007). *The Theory of Open Quantum Systems*. 1st ed. Oxford University PressOxford. DOI: 10.1093/acprof:oso/9780199213900.001.0001.

Burnett, Jonathan J. et al. (2019). "Decoherence benchmarking of superconducting qubits". *npj Quantum Information* 5.1, pp. 1–8. DOI: 10.1038/s41534-019-0168-5.

Cai, Zhenyu et al. (2023). "Quantum error mitigation". *Reviews of Modern Physics* 95.4, p. 045005. DOI: 10.1103/RevModPhys.95.045005.

Cattaneo, Marco, Gabriele De Chiara, Sabrina Maniscalco, Roberta Zambrini, and Gian Luca Giorgi (2021). "Collision Models Can Efficiently Simulate Any Multipartite Markovian Quantum Dynamics". *Physical Review Letters* 126.13, p. 130403. DOI: 10.1103/PhysRevLett.126.130403.

Chancellor, Nicholas, Aleks Kissinger, Joschka Roffe, Stefan Zohren, and Dominic Horsman (2023). *Graphical Structures for Design and Verification of Quantum Error Correction*. DOI: 10.48550/arXiv.1611.08012.

Choi, Man-Duen (1975). "Completely positive linear maps on complex matrices". *Linear Algebra and its Applications* 10.3, pp. 285–290. DOI: 10.1016/0024-3795(75)90075-0.

Chuang, Isaac L. and M. A. Nielsen (1997). "Prescription for experimental determination of the dynamics of a quantum black box". *Journal of Modern Optics* 44.11, pp. 2455–2467. DOI: 10.1080/09500349708231894.

Cirac, J. I. and P. Zoller (1995). "Quantum Computations with Cold Trapped Ions". *Physical Review Letters* 74.20, pp. 4091–4094. DOI: 10.1103/PhysRevLett.74.4091.

Cirstoiu, Cristina, Silas Dilkes, Daniel Mills, Seyon Sivarajah, and Ross Duncan (2023). "Volumetric Benchmarking of Error Mitigation with Qermit". *Quantum* 7, p. 1059. DOI: 10.22331/q-2023-07-13-1059.

Clarke, John and Frank K. Wilhelm (2008). "Superconducting quantum bits". *Nature* 453.7198, pp. 1031–1042. DOI: 10.1038/nature07128.

Conn, Andrew R., Nicholas I. M. Gould, and Philippe L. Toint (2000). *Trust-region methods*. MPS-SIAM series on optimization. Philadelphia, Pa: Society for Industrial and Applied Mathematics [u.a.] 959 pp.

Cross, Andrew W., Lev S. Bishop, Sarah Sheldon, Paul D. Nation, and Jay M. Gambetta (2019). "Validating quantum computers using randomized model circuits". *Physical Review A* 100.3, p. 032328. DOI: 10.1103/PhysRevA.100.032328.

Cross, Andrew W., Lev S. Bishop, John A. Smolin, and Jay M. Gambetta (2017). *Open Quantum Assembly Language*. DOI: 10.48550/arXiv.1707.03429.

Czarnik, Piotr, Andrew Arrasmith, Patrick J. Coles, and Lukasz Cincio (2021). "Error mitigation with Clifford quantum-circuit data". *Quantum* 5, p. 592. DOI: 10.22331/q-2021-11-26-592.

Dahlhauser, Megan L. and Travis S. Humble (2021). "Modeling noisy quantum circuits using experimental characterization". *Physical Review A* 103.4, p. 042603. DOI: 10.1103/PhysRevA.103.042603.

– (2022). *Benchmarking Characterization Methods for Noisy Quantum Circuits*. DOI: 10.48550/arXiv.2201.02243.

Daley, Andrew J. et al. (2022). "Practical quantum advantage in quantum simulation". *Nature* 607.7920, pp. 667–676. DOI: 10.1038/s41586-022-04940-6.

Deutsch, D. (1989). "Quantum Computational Networks". *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 425.1868, pp. 73–90.

Deutsch, David and Richard Jozsa (1997). "Rapid solution of problems by quantum computation". *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439.1907, pp. 553–558. DOI: 10.1098/rspa.1992.0167.

Devitt, Simon J., Ashley M. Stephens, William J. Munro, and Kae Nemoto (2013). "Requirements for fault-tolerant factoring on an atom-optics quantum computer". *Nature Communications* 4.1, p. 2524. DOI: 10.1038/ncomms3524.

Dirac, P. a. M. (1939). "A new notation for quantum mechanics". *Mathematical Proceedings of the Cambridge Philosophical Society* 35.3, pp. 416–418. DOI: 10.1017/S0305004100021162.

DiVincenzo, David P. (1995). "Two-bit gates are universal for quantum computation". *Physical Review A* 51.2, pp. 1015–1022. DOI: 10.1103/PhysRevA.51.1015.

– (2000). "The Physical Implementation of Quantum Computation". *Fortschritte der Physik* 48.9, pp. 771–783. DOI: https://doi.org/10.1002/1521-3978(200009)48:9/11<771::AID-PROP771>3.0.CO;2-E.

Emerson, Joseph et al. (2007). "Symmetrized Characterization of Noisy Quantum Processes". *Science* 317.5846, pp. 1893–1896. DOI: 10.1126/science.1145699.

Erhard, Alexander et al. (2019). "Characterizing large-scale quantum computers via cycle benchmarking". *Nature Communications* 10.1, p. 5347. DOI: 10.1038/s41467-019-13068-7.

Farhi, Edward, Jeffrey Goldstone, and Sam Gutmann (2014). *A Quantum Approximate Optimization Algorithm*. DOI: 10.48550/arXiv.1411.4028.

Farhi, Edward, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Daniel Preda (2001). "A Quantum Adiabatic Evolution Algorithm Applied to Random

Instances of an NP-Complete Problem". *Science* 292.5516, pp. 472–475. DOI: 10.1126/science.1057726.

Feynman, Richard P. (1982). "Simulating physics with computers". *International Journal of Theoretical Physics* 21.6, pp. 467–488. DOI: 10.1007/BF02650179.

Fogarty, M. A. et al. (2015). "Nonexponential fidelity decay in randomized benchmarking with low-frequency noise". *Physical Review A* 92.2, p. 022326. DOI: 10.1103/PhysRevA.92.022326.

Funcke, Lena, Tobias Hartung, Karl Jansen, Stefan Kühn, and Paolo Stornati (2021). "Dimensional Expressivity Analysis of Parametric Quantum Circuits". *Quantum* 5, p. 422. DOI: 10.22331/q-2021-03-29-422.

Funcke, Lena, Tobias Hartung, Karl Jansen, Stefan Kühn, Paolo Stornati, and Xiaoyang Wang (2022). "Measurement error mitigation in quantum computers through classical bit-flip correction". *Physical Review A* 105.6, p. 062404. DOI: 10.1103/PhysRevA.105.062404.

Gambetta, Jay M. et al. (2012). "Characterization of Addressability by Simultaneous Randomized Benchmarking". *Physical Review Letters* 109.24, p. 240504. DOI: 10.1103/PhysRevLett.109.240504.

Georgopoulos, Konstantinos, Clive Emary, and Paolo Zuliani (2021a). "Modeling and simulating the noisy behavior of near-term quantum computers". *Physical Review A* 104.6, p. 062432. DOI: 10.1103/PhysRevA.104.062432.

– (2021b). *Quantum Computer Benchmarking via Quantum Algorithms*. DOI: 10.48550/arXiv.2112.09457.

Gharibian, S. (2010). "Strong NP-hardness of the quantum separability problem". *Quantum Information and Computation* 10.3, pp. 343–360. DOI: 10.26421/QIC10.3-4-11.

Ghosh, Joydip, Andrei Galiautdinov, Zhongyuan Zhou, Alexander N. Korotkov, John M. Martinis, and Michael R. Geller (2013). "High-fidelity controlled-sigma^Z gate for resonator-based superconducting quantum computers". *Physical Review A* 87.2, p. 022309. DOI: 10.1103/PhysRevA.87.022309.

Gilchrist, Alexei, Nathan K. Langford, and Michael A. Nielsen (2005). "Distance measures to compare real and ideal quantum processes". *Physical Review A* 71.6, p. 062310. DOI: 10.1103/PhysRevA.71.062310.

Gray, Jim, ed. (1994). *The Benchmark handbook: for database and transaction processing systems*. 2. ed., 2. [print.] The Morgan Kaufman series in data management systems. San Francisco, Calif: Morgan Kaufmann. 592 pp.

Greenbaum, Daniel (2015). *Introduction to Quantum Gate Set Tomography*. DOI: 10.48550/arXiv.1509.02921.

Grover, Lov K. (1996). “A fast quantum mechanical algorithm for database search”. *Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing*. STOC ’96. New York, NY, USA: Association for Computing Machinery, pp. 212–219. DOI: 10.1145/237814.237866.

Gustiani, Cica, Tyson Jones, and Simon C. Benjamin (2023). *The Virtual Quantum Device (VQD): A tool for detailed emulation of quantum computers*. DOI: 10.48550/arXiv.2306.07342.

Hall, Brian C. (2013). *Quantum Theory for Mathematicians*. 1-566.

Harper, Robin, Steven T. Flammia, and Joel J. Wallman (2020). “Efficient learning of quantum noise”. *Nature Physics* 16.12, pp. 1184–1188. DOI: 10.1038/s41567-020-0992-8.

Hellinger, E. (1909). “Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen.” *Journal für die reine und angewandte Mathematik* 1909.136, pp. 210–271. DOI: 10.1515/crll.1909.136.210.

Helsen, Jonas, Xiao Xue, Lieven M. K. Vandersypen, and Stephanie Wehner (2019). “A new class of efficient randomized benchmarking protocols”. *npj Quantum Information* 5.1, pp. 1–9. DOI: 10.1038/s41534-019-0182-7.

Henriet, Loïc et al. (2020). “Quantum computing with neutral atoms”. *Quantum* 4, p. 327. DOI: 10.22331/q-2020-09-21-327.

Huppler, Karl (2009). “The Art of Building a Good Benchmark”. *Performance Evaluation and Benchmarking*. Ed. by Raghunath Nambiar and Meikel Poess. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 18–30. DOI: 10.1007/978-3-642-10424-4_3.

Isenberg, Tobias, Petra Isenberg, Jian Chen, Michael Sedlmair, and Torsten Möller (2013). “A Systematic Review on the Practice of Evaluating Visualization”. *IEEE Transactions on Visualization and Computer Graphics* 19.12, pp. 2818–2827. DOI: 10.1109/TVCG.2013.126.

Ishikawa, Kaoru, ed. (1976). *Guide to quality control*. Industrial engineering & technology. Tokyo: Asian Productivity Organization. 226 pp.

Jacobs, K. and P. L. Knight (1998). “Linear quantum trajectories: Applications to continuous projection measurements”. *Physical Review A* 57.4, pp. 2301–2310. DOI: 10.1103/PhysRevA.57.2301.

Jones, Tyson, Anna Brown, Ian Bush, and Simon C. Benjamin (2019). “QuEST and High Performance Simulation of Quantum Computers”. *Scientific Reports* 9.1, p. 10736. DOI: 10.1038/s41598-019-47174-9.

Jordan, P. and E. Wigner (1928). “Über das Paulische Äquivalenzverbot”. *Zeitschrift für Physik* 47.9, pp. 631–651. DOI: 10.1007/BF01331938.

Kadowaki, Tadashi and Hidetoshi Nishimori (1998). “Quantum annealing in the transverse Ising model”. *Physical Review E* 58.5, pp. 5355–5363. DOI: 10.1103/PhysRevE.58.5355.

Kantorovich, L. V. (1960). “Mathematical Methods of Organizing and Planning Production”. *Management Science* 6.4, pp. 366–422. DOI: 10.1287/mnsc.6.4.366.

Ketterer, Andreas and Thomas Wellens (2023). *Characterizing crosstalk of superconducting transmon processors*. DOI: 10.48550/arXiv.2303.14103.

Khazali, Mohammadsadegh and Klaus Mølmer (2020). “Fast Multiqubit Gates by Adiabatic Evolution in Interacting Excited-State Manifolds of Rydberg Atoms and Superconducting Circuits”. *Physical Review X* 10.2, p. 021054. DOI: 10.1103/PhysRevX.10.021054.

Kim, Dohun, D. R. Ward, et al. (2015). “Microwave-driven coherent operation of a semiconductor quantum dot charge qubit”. *Nature Nanotechnology* 10.3, pp. 243–247. DOI: 10.1038/nnano.2014.336.

Kim, Youngseok, Christopher J. Wood, et al. (2023). “Scalable error mitigation for noisy quantum circuits produces competitive expectation values”. *Nature Physics* 19.5, pp. 752–759. DOI: 10.1038/s41567-022-01914-3.

Kimmel, Shelby, Guang Hao Low, and Theodore J. Yoder (2015). “Robust calibration of a universal single-qubit gate set via robust phase estimation”. *Physical Review A* 92.6, p. 062315. DOI: 10.1103/PhysRevA.92.062315.

Kistowski, Jóakim v., Jeremy A. Arnold, Karl Huppler, Klaus-Dieter Lange, John L. Henning, and Paul Cao (2015). “How to Build a Benchmark”. *Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering*. ICPE ’15. New York, NY, USA: Association for Computing Machinery, pp. 333–336. DOI: 10.1145/2668930.2688819.

Kitaev, A. Yu. (2003). “Fault-tolerant quantum computation by anyons”. *Annals of Physics* 303.1, pp. 2–30. DOI: 10.1016/S0003-4916(02)00018-0.

Klimov, P. V. et al. (2018). “Fluctuations of Energy-Relaxation Times in Superconducting Qubits”. *Physical Review Letters* 121.9, p. 090502. DOI: 10.1103/PhysRevLett.121.090502.

Knapp, Christina, Michael Beverland, Dmitry I. Pikulin, and Torsten Karzig (2018). “Modeling noise and error correction for Majorana-based quantum computing”. *Quantum* 2, p. 88. DOI: 10.22331/q-2018-09-03-88.

Knill, E., R. Laflamme, and G. J. Milburn (2001). “A scheme for efficient quantum computation with linear optics”. *Nature* 409.6816, pp. 46–52. DOI: 10.1038/35051009.

Knill, E., D. Leibfried, et al. (2008). “Randomized benchmarking of quantum gates”. *Physical Review A* 77.1, p. 012307. DOI: 10.1103/PhysRevA.77.012307.

Knill, Emanuel, Raymond Laflamme, and Wojciech H. Zurek (1998). “Resilient Quantum Computation”. *Science* 279.5349, pp. 342–345. DOI: 10.1126/science.279.5349.342.

Kounev, Samuel (2021). *Systems Benchmarking: For scientists and engineers.* S.l.: SPRINGER.

Kraus, Karl, A. Böhm, J. D. Dollard, and W. H. Wootters, eds. (1983). *States, Effects, and Operations Fundamental Notions of Quantum Theory*. Vol. 190. Lecture Notes in Physics. Berlin, Heidelberg: Springer. DOI: 10.1007/3-540-12732-1.

Kullback, S. and R. A. Leibler (1951). “On Information and Sufficiency”. *The Annals of Mathematical Statistics* 22.1, pp. 79–86. DOI: 10.1214/aoms/1177729694.

Kwon, Hyeokjea and Joonwoo Bae (2021). “A Hybrid Quantum-Classical Approach to Mitigating Measurement Errors in Quantum Algorithms”. *IEEE Transactions on Computers* 70.9, pp. 1401–1411. DOI: 10.1109/TC.2020.3009664.

Li, Boxi, Shahnawaz Ahmed, et al. (2022). “Pulse-level noisy quantum circuits with QuTiP”. *Quantum* 6, p. 630. DOI: 10.22331/q-2022-01-24-630.

Li, Ying and Simon C. Benjamin (2017). “Efficient Variational Quantum Simulator Incorporating Active Error Minimization”. *Physical Review X* 7.2, p. 021050. DOI: 10.1103/PhysRevX.7.021050.

Lindblad, G. (1976). “On the generators of quantum dynamical semigroups”. *Communications in Mathematical Physics* 48.2, pp. 119–130. DOI: 10.1007/BF01608499.

Lubinski, Thomas et al. (2023). “Application-Oriented Performance Benchmarks for Quantum Computing”. *IEEE Transactions on Quantum Engineering* 4, pp. 1–32. DOI: 10.1109/TQE.2023.3253761.

Majumder, Swarnadeep et al. (2023). “Characterizing and mitigating coherent errors in a trapped ion quantum processor using hidden inverses”. *Quantum* 7, p. 1006. DOI: 10.22331/q-2023-05-15-1006.

Merino, L., M. Ghafari, C. Anslow, and O. Nierstrasz (2018). "A systematic literature review of software visualization evaluation". *Journal of Systems and Software* 144, pp. 165–180. DOI: 10.1016/j.jss.2018.06.027.

Merkel, Seth T. et al. (2013). "Self-consistent quantum process tomography". *Physical Review A* 87.6, p. 062119. DOI: 10.1103/PhysRevA.87.062119.

Mitchell, Melanie (1996). *An introduction to genetic algorithms*. Complex adaptive systems. Cambridge, Mass: MIT Press. 205 pp.

Nielsen, Erik, John King Gamble, Kenneth Rudinger, Travis Scholten, Kevin Young, and Robin Blume-Kohout (2021). "Gate Set Tomography". *Quantum* 5, p. 557. DOI: 10.22331/q-2021-10-05-557.

Nielsen, Michael A. (2003). "Quantum computation by measurement and quantum memory". *Physics Letters A* 308.2, pp. 96–100. DOI: 10.1016/S0375-9601(02)01803-0.

Nielsen, Michael A. and Isaac L. Chuang (2010). *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Higher Education from Cambridge University Press. DOI: 10.1017/CBO9780511976667. (Visited on 06/07/2023).

Paladino, E., A. D'Arrigo, A. Mastellone, and G. Falci (2011). "Decoherence times of universal two-qubit gates in the presence of broad-band noise". *New Journal of Physics* 13.9, p. 093037. DOI: 10.1088/1367-2630/13/9/093037.

Pearl, Judea (2000). *Causality: models, reasoning, and inference*. Cambridge, U.K. ; New York: Cambridge University Press. 384 pp.

Peruzzo, Alberto et al. (2014). "A variational eigenvalue solver on a photonic quantum processor". *Nature Communications* 5.2, pp. 1–10. DOI: 10.1038/ncomms5213.

Plenio, M. B. and P. L. Knight (1998). "The quantum-jump approach to dissipative dynamics in quantum optics". *Reviews of Modern Physics* 70.1, pp. 101–144. DOI: 10.1103/RevModPhys.70.101.

Powell, M. J. D. (1970). "A New Algorithm for Unconstrained Optimization". *Nonlinear Programming*. Ed. by J. B. Rosen, O. L. Mangasarian, and K. Ritter. Academic Press, pp. 31–65. DOI: 10.1016/B978-0-12-597050-1.50006-3.

Poyatos, J. F., J. I. Cirac, and P. Zoller (1997). "Complete Characterization of a Quantum Process: The Two-Bit Quantum Gate". *Physical Review Letters* 78.2, pp. 390–393. DOI: 10.1103/PhysRevLett.78.390.

Preskill, John (2018). “Quantum computing in the NISQ era and beyond”. *Quantum* 2 (July), pp. 1–20. DOI: 10.22331/q-2018-08-06-79.

Proctor, Timothy, Melissa Revelle, et al. (2020). “Detecting and tracking drift in quantum information processors”. *Nature Communications* 11.1, p. 5396. DOI: 10.1038/s41467-020-19074-4.

Proctor, Timothy, Kenneth Rudinger, Kevin Young, Erik Nielsen, and Robin Blume-Kohout (2022). “Measuring the capabilities of quantum computers”. *Nature Physics* 18.1, pp. 75–79. DOI: 10.1038/s41567-021-01409-7.

Proctor, Timothy, Kenneth Rudinger, Kevin Young, Mohan Sarovar, and Robin Blume-Kohout (2017). “What Randomized Benchmarking Actually Measures”. *Physical Review Letters* 119.13, p. 130502. DOI: 10.1103/PhysRevLett.119.130502.

Proctor, Timothy J., Arnaud Carignan-Dugas, Kenneth Rudinger, Erik Nielsen, Robin Blume-Kohout, and Kevin Young (2019). “Direct Randomized Benchmarking for Multi-qubit Devices”. *Physical Review Letters* 123.3, p. 030503. DOI: 10.1103/PhysRevLett.123.030503.

Qiskit contributors (2023). *Qiskit: An Open-source Framework for Quantum Computing*. DOI: 10.5281/zenodo.2573505.

Raussendorf, Robert and Hans J. Briegel (2001). “A One-Way Quantum Computer”. *Physical Review Letters* 86.22, pp. 5188–5191. DOI: 10.1103/PhysRevLett.86.5188.

Resch, Salonik and Ulya R. Karpuzcu (2021). “Benchmarking Quantum Computers and the Impact of Quantum Noise”. *ACM Computing Surveys* 54.7, 142:1–142:35. DOI: 10.1145/3464420.

Ruan, Shaolun, Yong Wang, Weiwen Jiang, Ying Mao, and Qiang Guan (2023). “VACSEN: A Visualization Approach for Noise Awareness in Quantum Computing”. *IEEE Transactions on Visualization and Computer Graphics* 29.1, pp. 462–472. DOI: 10.1109/TVCG.2022.3209455.

Rudinger, Kenneth et al. (2021). “Experimental Characterization of Crosstalk Errors with Simultaneous Gate Set Tomography”. *PRX Quantum* 2.4, p. 040338. DOI: 10.1103/PRXQuantum.2.040338.

Sarovar, Mohan, Timothy Proctor, Kenneth Rudinger, Kevin Young, Erik Nielsen, and Robin Blume-Kohout (2020). “Detecting crosstalk errors in quantum information processors”. *Quantum* 4, p. 321. DOI: 10.22331/q-2020-09-11-321.

Sedlmair, Michael, Miriah Meyer, and Tamara Munzner (2012). "Design Study Methodology: Reflections from the Trenches and the Stacks". *IEEE Transactions on Visualization and Computer Graphics* 18.12, pp. 2431–2440. DOI: 10.1109/TVCG.2012.213.

Shor, P.W. (1996). "Fault-tolerant quantum computation". *Proceedings of 37th Conference on Foundations of Computer Science*. Proceedings of 37th Conference on Foundations of Computer Science, pp. 56–65. DOI: 10.1109/SFCS.1996.548464.

Shor, Peter W. (1997). "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". *SIAM Journal on Computing* 26.5, pp. 1484–1509. DOI: 10.1137/S0097539795293172.

Silva, Adenilton José da, Teresa Bernarda Ludermir, and Wilson Rosa de Oliveira (2016). "Quantum perceptron over a field and neural network architecture selection in a quantum computer". *Neural Networks* 76, pp. 55–64. DOI: 10.1016/j.neunet.2016.01.002.

Sim, S.E., S. Easterbrook, and R.C. Holt (2003). "Using benchmarking to advance research: a challenge to software engineering". *25th International Conference on Software Engineering, 2003. Proceedings.* 25th International Conference on Software Engineering, 2003. Proceedings. Pp. 74–83. DOI: 10.1109/ICSE.2003.1201189.

Skadron, K., M. Martonosi, D.I. August, M.D. Hill, D.J. Lilja, and V.S. Pai (2003). "Challenges in computer architecture evaluation". *Computer* 36.8, pp. 30–36. DOI: 10.1109/MC.2003.1220579.

Spall, J.C. (2000). "Adaptive stochastic approximation by the simultaneous perturbation method". *IEEE Transactions on Automatic Control* 45.10, pp. 1839–1853. DOI: 10.1109/TAC.2000.880982.

Spall, James C. (1992). "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation". *IEEE Transactions on Automatic Control* 37.3, pp. 332–341. DOI: 10.1109/9.119632.

Takagi, Ryuji, Suguru Endo, Shintaro Minagawa, and Mile Gu (2022). "Fundamental limits of quantum error mitigation". *npj Quantum Information* 8.1, pp. 1–11. DOI: 10.1038/s41534-022-00618-z.

Tannu, Swamit S. and Moinuddin K. Qureshi (2019). "Mitigating Measurement Errors in Quantum Computers by Exploiting State-Dependent Bias". *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*. MICRO '52. New York, NY, USA: Association for Computing Machinery, pp. 279–290. DOI: 10.1145/3352460.3358265.

Temme, Kristan, Sergey Bravyi, and Jay M. Gambetta (2017). “Error Mitigation for Short-Depth Quantum Circuits”. *Physical Review Letters* 119.18, pp. 1–15. DOI: 10.1103/PhysRevLett.119.180509.

Trout, Colin J. et al. (2018). “Simulating the performance of a distance-3 surface code in a linear ion trap”. *New Journal of Physics* 20.4, p. 043038. DOI: 10.1088/1367-2630/aab341.

Turchette, Q. A., C. J. Hood, W. Lange, H. Mabuchi, and H. J. Kimble (1995). “Measurement of Conditional Phase Shifts for Quantum Logic”. *Physical Review Letters* 75.25, pp. 4710–4713. DOI: 10.1103/PhysRevLett.75.4710.

Virtanen, Pauli et al. (2020). “SciPy 1.0: fundamental algorithms for scientific computing in Python”. *Nature Methods* 17.3, pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

Vogel, K. and H. Risken (1989). “Determination of quasiprobability distributions in terms of probability distributions for the rotated quadrature phase”. *Physical Review A* 40.5, pp. 2847–2849. DOI: 10.1103/PhysRevA.40.2847.

Wallman, Joel J. and Joseph Emerson (2016). “Noise tailoring for scalable quantum computation via randomized compiling”. *Physical Review A* 94.5, p. 052325. DOI: 10.1103/PhysRevA.94.052325.

Weber, Tom, Kerstin Borras, Karl Jansen, Dirk Krücker, and Matthias Riebisch (2024). “Construction and volumetric benchmarking of quantum computing noise models”. *Physica Scripta* 99.6, p. 065106. DOI: 10.1088/1402-4896/ad406c.

Weber, Tom, Matthias Riebisch, Kerstin Borras, Karl Jansen, and Dirk Krücker (2021). “Modelling for Quantum Error Mitigation”. *2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C)*. 2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C), pp. 102–105. DOI: 10.1109/ICSA-C52384.2021.00026.

Wilde, Mark M. (2017). *Quantum Information Theory*. 2nd ed. Cambridge: Cambridge University Press. DOI: 10.1017/9781316809976.

Williams, Colin P. (2011). *Explorations in Quantum Computing*. Texts in Computer Science. London: Springer. DOI: 10.1007/978-1-84628-887-6.

Wu, Yulin et al. (2021). “Strong Quantum Computational Advantage Using a Superconducting Quantum Processor”. *Physical Review Letters* 127.18, p. 180501. DOI: 10.1103/PhysRevLett.127.180501.

Yao, Andrew Chi-Chih (1993). “Quantum Circuit Complexity”. *IEEE Annual Symposium on Foundations of Computer Science*.

Ying, Mingsheng (2021). “Model Checking for Verification of Quantum Circuits”. *arXiv*, pp. 1–16.

Zhong, Han-Sen et al. (2020). “Quantum computational advantage using photons”. *Science* 370.6523, pp. 1460–1463. DOI: `10.1126/science.abe8770`.

GLOSSARY

CPTP completely positive, trace-preserving. 23, 24, 34, 42, 48, 49, 120

DAG directed acyclic graph. 46, 48, 49, 51, 55, 111

HD Hellinger distance. 88, 91, 94, 95, 108

KL Kullback-Leibler. 66, 88, 91, 93, 95, 96, 108, 120

LS least-squares. 90, 91, 95

NISQ Noisy Intermediate-Scale Quantum. 1, 15, 18, 20, 23, 43, 64, 67, 80, 82

NQC noisy quantum circuit. 42, 45, 49–55, 58, 111

QAOA quantum approximate optimization algorithm. 53, 64

SPAM state preparation and measurement. 33, 34, 42

SPSA simultaneous perturbation stochastic approximation. 35, 90, 91, 93, 94, 96, 99, 107, 120, 121

VQA variational quantum algorithm. 58, 62, 64, 65, 67, 75, 78, 88, 90

VQE Variational Quantum Eigensolver. 15, 18, 53