Construction and volumetric benchmarking of quantum computing noise models

Physica Scripta

PAPER

Construction and volumetric benchmarking of quantum computing noise models

Tom Weber[1], **Kerstin Borras**[2,3], **Karl Jansen**[4,5], **Dirk Krücker**[2] and **Matthias Riebisch**[1,†]

[1] Department of Informatics, University of Hamburg, Germany
[2] Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany
[3] RWTH Aachen University, Aachen, Germany
[4] CQTA, Deutsches Elektronen-Synchrotron DESY, Zeuthen, Germany
[5] Computation-Based Science and Technology Research Center, The Cyprus Institute, Cyprus
[†] Deceased.

E-mail: tom.weber-1@uni-hamburg.de

Keywords: quantum computing, noise model, volumetric benchmark, parameter optimization

OPEN ACCESS

RECEIVED
19 September 2023

REVISED
8 April 2024

ACCEPTED FOR PUBLICATION
18 April 2024

PUBLISHED
3 May 2024

Abstract

The main challenge of quantum computing on its way to scalability is the erroneous behaviour of current devices. Understanding and predicting their impact on computations is essential to counteract these errors with methods such as quantum error mitigation. Thus, it is necessary to construct and evaluate accurate noise models. However, the evaluation of noise models does not yet follow a systematic approach, making it nearly impossible to estimate the accuracy of a model for a given application. Therefore, we developed and present a systematic approach to benchmarking noise models for quantum computing applications. It compares the results of hardware experiments to predictions of noise models for a representative set of quantum circuits. We also construct a noise model containing five types of quantum noise and optimize its parameters using a series of training circuits. We compare its accuracy to other noise models by volumetric benchmarks involving typical variational quantum circuits. The model can easily be expanded by adding new quantum channels.

1. Introduction

Quantum computing is expected to offer novel applications in numerous fields of science. The most significant challenge to achieving scalable quantum computing is the level of errors in current noisy intermediate-scale quantum (NISQ) devices [1]. Counteracting these errors is essential to enable reliable computations. While potential prospect solutions such as quantum error correction remain impracticable [2] due to small qubit numbers, quantum error mitigation methods can improve results significantly without causing an overhead of necessary qubit resources. Various methods aim to tackle different types of errors for several applications. Especially for algorithms like Variational Quantum Eigensolver (VQE) [3], we already find a large number of protocols trying to mitigate, e.g., readout error, gate error, or cross-talk [4–12]. Almost all error mitigation methods have in common that they require additional quantum computing time [13], which is limited. A prioritization of the dominant types of error is therefore necessary. Understanding and predicting the noisy behaviour of a quantum computer makes accurate noise models are indispensable for efficient and reliable quantum computing calculations.

However, there is no systematic approach for evaluating the quality of a noise model. Its accuracy is often estimated using a small number of arbitrary test circuits and comparing the models prediction to the results obtained with quantum hardware. These test circuits are usually not similar enough to realistic application circuits to allow for a generalization of the results. The size of the quantum circuits is usually too small, making it difficult to assess the accuracy of a noise model in a realistic application context.

Therefore, we propose a volumetric benchmarking approach that systematically evaluates noise models, and present a technique to optimize parameters of a noise model we construct in this work. The benchmarks are

based on the framework presented in [14], which measures the performance of quantum computers. They compare the predictions of a noise model to results from quantum hardware for a choice of representative quantum circuits. This procedure is carried out for different pairs (w,d) of width w and depth d of the circuits, where the width corresponds to the number of qubits, and the depth can be related to the number of consecutive gates or layers thereof. This allows evaluating the quality of a noise model as a function of the problem complexity, hence the name volumetric benchmark. The applications of our benchmarks are versatile. With a noise model assessed as being accurate, one can use simulations employing this model for test purposes or if quantum resources are limited. The knowledge gained through the model can also help prioritize quantum error mitigation methods.

The noise model constructed in section 4 depends on a set of trainable parameters for the specific kind of noise included in the model and aims to represent the noise impact on VQE or similar algorithms. The parameters correspond to probabilities that certain errors occur and are optimized using the SPSA [15] algorithm. The noise model is easily expandable by additional noise channels and the parameter optimization can be used for all types of quantum circuits and included types of noise. We then conduct volumetric benchmarks of the resulting noise model. The hardware experiments are run on IBM quantum hardware.

1.1. Contribution

This paper contains two main contributions:

1. A benchmarking protocol for measuring the accuracy of quantum computing noise models, including a discussion regarding quality attributes from the systems benchmarking literature.
2. The construction of a noise model, training of its parameters, and an evaluation with volumetric benchmarks using typical VQE quantum circuits on IBM quantum hardware. The model is compared to the `ibmq_manila` device noise model provided in qiskit [16].

1.2. Related work

This section summarizes relevant research related to this paper. It discusses the literature on benchmarking for quantum hardware, the construction of noise models, and their calibration.

A variety of tomography approaches exists for characterizing and benchmarking quantum computers [17]. Quantum State Tomography (QST) [18] is a procedure that characterizes an unknown state ρ. This state could then be compared to the expected, ideal state of a quantum computation to obtain an estimate of the fidelity of the hardware. Quantum Process Tomography (QPT) [19–24] measures the process matrix of quantum gates. Both QST and QPT consider state preparation and measurements to work correctly. This is presently not always the case, making the estimates of quantum states and gates erroneous. In contrast, our noise model includes state preparation and measurement (SPAM) errors, and our training approach ensures they are represented appropriately. Gate Set Tomography (GST) [25–27] also takes into account erroneous state preparation and measurements. While QPT characterizes a single gate, GST can reconstruct a set of operations in a self-consistent way. Many quantum experiments are needed to achieve this characterization, and scalability is problematic for benchmarking large systems. QPT and GST attempt to describe the noisy processes of a quantum device, but the process matrices do not give conceptual insights into the errors. While first approaches have been presented to interpret these process matrices [28], this paper comprehensively constructs a noise model derived from the underlying physical processes.

Randomized benchmarking (RB) [29–31] measures the average gate error rates of a quantum computer. Many variations of RB exist, including cycle benchmarking [32]. These methods evaluate the performance of quantum hardware and do not attempt to describe the noise in detail. Moreover, they do not provide prospects on the impact of the noise.

In addition to all these tomography methods, there are other prominent benchmarking approaches for quantum hardware. Quantum volume [33] is a single-number metric that indicates the maximal size of quantum circuits that can be executed successfully on a device. In [14], the authors propose a volumetric benchmarking approach that generalizes the quantum volume metric. Volumetric benchmarks using mirror circuits are conducted in [34] and applied to quantum error mitigation in [35]. Our work transfers volumetric benchmarks to a different setting. While they originally measure the capabilities of quantum hardware for different problem sizes, we evaluate the accuracy of noise models in a volumetric manner and aim to provide prospects on the potential impact of that noise.

In [36], the authors present a wildcard error that accounts for inconsistencies between noise model predictions and hardware data. The amount of wildcard error needed can be interpreted as an estimate of the accuracy of the noise model. Machine learning methods are also used to describe quantum noise. In [37], the authors propose a learning procedure to obtain the error rates of a quantum computer. In [38], a noise model

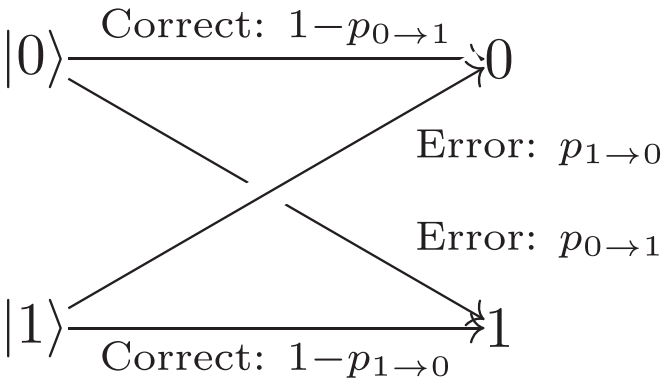


Figure 1. Graphical representation of readout error on a single qubit with different bit-flip probabilities $p_{0\to1}$ and $p_{1\to0}$.

construction and its evaluation with test quantum circuits are presented. Our noise model adds crosstalk and a more advanced representation of readout error. Furthermore, we introduce a more systematic approach for benchmarking the noise models and present a different training approach for the model parameters.

2. Background

This section introduces the different types of noise that can occur on a quantum computer, our notation, and quality criteria for benchmarks.

2.1. Quantum noise and noise models

Quantum noise refers to all interactions of a quantum system with its environment. In quantum computing, these interactions lead to erroneous computations. Not only do qubits interact with their environment, but also with each other. Therefore, quantum circuits are not executed as intended. For instance, measurements can be faulty (readout error), or gates are applied imperfectly (gate error).

Noise models offer means of describing and predicting the noisy effects in a quantum device. They contain information about the error types and the point in a quantum circuit where they occur. More precisely, a noise model maps quantum circuits to outcome probability distributions [36]. One would obtain these distributions by running the circuits many times on a noisy quantum computer that behaves the way the model describes.

Quantum operations [39] on density matrices (or density operators) are the prevalent mathematical formalism to model quantum noise. This section does not give detailed mathematical definitions of quantum operations but focuses on specific examples of noise channels.We refer to appendices A and B instead for more mathematical details. A comprehensive discussion of the subject can also be found in [39].

In the following, let ρ be a density matrix describing the state of a set of qubits and denote the Pauli matrices as

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

2.1.1. Readout error

The measurement of qubits on current quantum computers is often erroneous, with error rates of up to 30% [40], although this has improved on the newest machines. This behaviour is called measurement error or readout error and can be modelled as a classical bit-flip as follows [5]: For each qubit q, a measurement outcome '0' is mistakenly recorded as '1' with probability $p_{0\to1}(q)$ and vice versa with $p_{1\to0}(q)$ (as shown in figure 1). Note that the probabilities can be asymmetric. In this work, we use the terms readout error and measurement error interchangeably.

Several papers have been published on readout errors and methods to mitigate them, particularly for measuring expectation values of observables on a quantum computer. Obtaining these expectation values is an essential step in VQE algorithms [3].

2.1.2. State preparation error

At the beginning of a quantum circuit, the qubits of a quantum computer are prepared in an initial state. Typically, the state $\rho_0 = |0\cdots0\rangle\langle0\cdots0|$ is chosen, where $|0\cdots0\rangle = |0\rangle^{\otimes N}$ and N denotes the number of qubits.

This procedure can be imperfect, resulting in an incorrect initial state and, thereby, unreliable computation with low fidelity.

When the initial state is given as above, the state preparation error can be modelled as applying an X gate on qubit q with probability $p_{\text{sp}}(q)$ [41]. For a single qubit, this yields the following noise channel:

$$\rho_0 \mapsto (1 - p_{\text{sp}}(q))\rho_0 + p_{\text{sp}}(q) \cdot \mathsf{X}\rho_0\mathsf{X}. \tag{1}$$

2.1.3. Depolarizing error

Besides state preparation and measurement, gate operations are imperfect. On current devices, both one and two-qubit gates are affected, where the error rates of two-qubit gates are usually higher [42].

Depolarizing error is an important type of gate error. A qubit is depolarized if its state is completely mixed and all information is lost. In terms of density matrices, the state ρ is replaced by the normalized identity matrix I/D with probability λ, where D is the dimension of the quantum system, i.e., $D = 2^N$ for depolarization of N qubits. The depolarizing channel can be written as

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

Following [39], one can rewrite the equation above for a single qubit as

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

If depolarization affects a gate g on a single qubit q (or a pair of qubits q_1, q_2), we denote the probability by $\lambda_g(q)$ (or $\lambda_g(q_1, q_2)$).

2.1.4. Thermal relaxation and dephasing

As a qubit interacts with its environment, it is subject to two central dynamics: thermal relaxation towards its ground state and dephasing [38, 43]. Assuming that the qubit is realized with $|0\rangle$ as its energetic ground state, thermal relaxation refers to the decay towards $|0\rangle$ over time. The mean lifetime of that decay is commonly labelled T_1.

Moreover, a qubit experiences a decay towards classical behaviour called dephasing. Similarly to thermal relaxation, this decay is determined by the time T_2. The times T_1 and T_2 are related by $T_2 \leqslant 2 \cdot T_1$.

For simplicity, we first assume $T_2 < T_1$. In that case, thermal relaxation can be modelled as a reset operator $|0\rangle\langle 0|$ that acts on the density matrix ρ with probability p_{reset} [38]. During quantum computation, the probability depends on the time T_g it takes to apply a gate operation g to the qubits. It is given by $p_{\text{reset}} = 1 - \exp(-T_g/T_1)$.

Dephasing can be modelled as the Pauli Z operator acting with probability p_{Z}. This probability is computed from the times T_1, T_2, and T_g by [38]

$$p_{\mathsf{Z}} = \frac{(1 - p_{\text{reset}})(1 - \exp(-T_g/T_2 + T_g/T_1))}{2}.$$

Thus, the noise channel representing thermal relaxation and dephasing can be written as

$$\mathcal{T}(\rho) = p_{\mathsf{I}}\rho + p_{\mathsf{Z}} \cdot \mathsf{Z}\rho\mathsf{Z} + p_{\text{reset}} \cdot |0\rangle\langle 0|\rho|0\rangle\langle 0|, \tag{3}$$

where $p_{\mathsf{I}} = 1 - p_{\mathsf{Z}} - p_{\text{reset}}$. If $T_2 > T_1$, one cannot write thermal relaxation and dephasing as above but must switch to a representation by a Choi matrix [44] instead. A detailed discussion can be found in [38].

In our situation, we write $T_{1,2}(q)$ for the $T_{1,2}$ time corresponding to qubit q. For a two-qubit operation on qubits q_1 and q_2, thermal relaxation and dephasing are considered to be two instances of single-qubit thermal relaxation and dephasing with the respective parameters $T_{1,2}(q_1)$ and $T_{1,2}(q_2)$.

2.1.5. Crosstalk error

The error types discussed above consider the interactions of a qubit with its environment to be local and independent of other qubits. In reality, many processes violate locality or independence. These processes are called crosstalk and lead to crosstalk errors. In this paper, we only consider a basic model of crosstalk error. An extensive discussion can be found in [45].

We represent crosstalk error as follows. Each time an erroneous single-qubit gate $g \in \{\mathsf{X}, \sqrt{\mathsf{X}}\}$ is applied to qubit q, it causes a rotation

$$\mathsf{R}_x(\phi) = \exp\left(-i\frac{\phi_g(q)}{2}\mathsf{X}\right) \tag{4}$$

on its neighbour qubits, where $\phi_g(q)$ is the rotation parameter.

2.2. Benchmarking quality criteria

Comprehensive literature exists on benchmarking classical computing systems or components. In the following, we review the most important aspects relevant to this paper. In [46], a benchmark is defined as a tool for evaluating or comparing systems according to specific characteristics. These characteristics can be assigned to one of these categories of *quality criteria*:

- **Relevance**: A benchmark is relevant if it measures the behaviour of a system well, and its results can be generalized to real-world scenarios. In our situation, a benchmark should allow an estimation of the accuracy of a given noise model for applications of interest.
- **Reproducibility**: The results of a reproducible benchmark are consistent over multiple runs with the same configuration.
- **Fairness**: A benchmark is fair if it does not impose artificial constraints on the system under test. Hence, a benchmark for noise models should not favour one model over another *a priori*.
- **Verifiability**: The verifiability of a benchmark ensures it is performed correctly and instructions are respected. One measure of improving verifiability is self-validation.
- **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.

3. Volumetric benchmarks for noise models

In this section, we present our volumetric benchmarking approach for evaluating the accuracy of noise models. First, we explain this process in detail and discuss the differences to the framework explained in [14]. Afterwards, we describe how improvements in the quality of such benchmarks in terms of the quality attributes from section 2.2 can be achieved.

3.1. The Framework

A volumetric benchmark in our approach is always related to a noise model and a quantum device. For a collection of test circuits, it compares the model predictions to the results of the quantum device. A volumetric benchmark consists of the following steps:

1. Test circuits: For pairs of width w and depth d, define a set $C(w, d)$ of quantum circuits. These circuits are used to compare the results predicted by the noise model to hardware results from the device. The depth could correspond to the number of gates or the number of layers thereof, while the width w is the number of qubits.
2. Compilation rules: Set up rules for compiling the quantum circuits from step 1 to the native gates of the device, enabling it to run the circuits later. There are different methods to compile quantum circuits. Sometimes, it is more feasible to optimize the circuits during compilation. In other cases, one might be more interested in rules restricting this optimization.
3. Model predictions: Specify a way to obtain the noise models predictions for the compiled quantum circuits. Among other things, such predictions could be made from noisy simulations of the circuits or exact computations using density matrices. We want to emphasize that the noise model must predict the results for the compiled circuits to allow for a meaningful comparison to hardware results. Further details are discussed later.
4. Hardware results: Run the compiled quantum circuits on the quantum computer. The exact specifications of this run, e.g., order of execution or number of shots, need to be described in detail for better reproducibility.
5. Single circuit evaluation: Define a metric that measures the difference between model prediction and hardware results for a single quantum circuit. For example, this metric could directly compare the outcome distributions of the circuit, or it could be based on higher-level attributes like expectation values of quantum mechanical observables.
6. Overall evaluation: If the set $C(w, d)$ contains more than a single quantum circuit, specify how to derive an overall evaluation. For example, when a single circuit is assessed using the difference in observable expectations, this overall evaluation could be chosen as the average difference.

While the approach presented in [14] compares hardware results to the ideal outcomes of quantum circuits, our approach uses hardware results and noise model predictions. Therefore, the third step is a novel extension of the original volumetric benchmarks. Moreover, our goal here is entirely different as we do not evaluate the behaviour of quantum devices but the accuracy of noise models.

Various attributes exist that can be used to classify quantum circuits. Two prominent attributes are their expressive power and the degree of entanglement. They are particularly important for VQE because the ground state cannot be found if the ansatz circuits are not sufficiently expressive. Both properties can be derived directly from the volume (w,d) in case of the `EfficientSU2` circuits that are used later in this work, see [47].

3.2. Ensuring quality

In section 2.2, we discuss benchmarking quality criteria such as relevance or reproducibility. The quality of our volumetric benchmarking framework for noise models in terms of these criteria depends on user choices during the different steps of a benchmark. This section discusses possibilities to increase the quality by examining the criteria individually. We first highlight potential obstacles for each attribute and explain how to avoid or minimize these.

3.2.1. Relevance

Often, relevance is the essential quality attribute of a benchmark. Even if it perfectly meets all other attributes perfectly, it can be useless because its results are not transferrable to any real-life application of interest. For example, a noise model might perform well in a benchmark but be inaccurate at predicting the noisy hardware behaviour when used for an application such as VQE.

The relevance of our volumetric benchmarks strongly depends on the quantum circuits and evaluation criteria used. Typical quantum circuits should be chosen if one is interested in noise models for a particular field of application. Moreover, the more quantum circuits are used and the more they differ from each other, the more transferrable the benchmark results are. For parametrizable quantum circuits, this means that several sets of parameters should be used to avoid a dependency on a particular choice. The scalability can be increased by testing many configurations (w,d) of width and depth.

3.2.2. Reproducibility

Quantum computations and, thus, also volumetric benchmarks are naturally subject to fluctuations due to finite shot numbers. The consequence is that reproducibility can suffer because running the same test circuits can yield different results. Since quantum hardware also often exhibits a time drift, its performance depends on when it is used. Hardware calibrations can have a significant impact on the noise of a device.

All benchmark experiments should be run in as small a time window as possible to mitigate these effects. Randomizing the circuit order could also reduce the impact of drift. Moreover, the circuits should be run as often as possible to decrease the variance of their outcome. If the model predictions are obtained by simulations, they should also be performed with large shot numbers. Alternatively, one could use exact predictions based on density matrix computations to minimize fluctuations further.

3.2.3. Fairness

Since our approach aims to benchmark noise models, artificial constraints on their performance are unlikely. If two noise models are supposed to predict the hardware behaviour for entirely different application contexts, they should not be compared in the first place. Fairness is mainly threatened when models are benchmarked at different times, and the hardware shows different levels of noise such that converging predictions can be more challenging. Again, this can be mitigated by performing benchmarks in a short time window and more often.

3.2.4. Verifiability

The verifiability of a benchmark measures to what extent it runs as expected (see [46]). Optimally, a verifiable benchmark includes some self-validation. For quantum circuits run on noisy hardware, such self-validation is challenging to achieve.

One could run simple circuits of which the ideal results are known and ensure that the hardware results are within a reasonable deviation range. Mirror circuits provide a more complex alternative because they have a richer structure while retaining easily predictable results [34]. The model predictions could be validated by comparing exact computations to noisy simulations.

3.2.5. Usability

The main threat to usability is restricted access to quantum hardware, so not everyone can easily perform a volumetric benchmark. Publicly providing quantum experiments data can help researchers run benchmarks

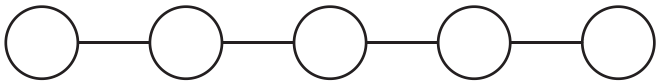


Figure 2. Qubit layout for IBM's `ibmq_manila` device.

with that data to test their noise models. More generally, the benchmarking software should be easy to use. This is not specific to benchmarks for quantum computing noise models but holds for all benchmarks. The circuit transpilation also impacts the usability of a benchmark because the user must be in full control over this process to ensure the reproducibility and other attributes.

4. Methods

This section describes the construction, training, and evaluation of a noise model for quantum computing. The noise model incorporates erroneous state preparation and measurement, depolarization, thermal relaxation, and crosstalk error. It is inspired by the model provided in [38] and depends on a set of parameters determined by training the model in a machine learning-like fashion. A detailed introduction of the noise model and the parameter optimization is given later.

This section is structured as follows. Firstly, we construct the noise model without specifying its parameters. It is defined on the native gate set of the IBM Quantum Falcon processors. Secondly, we explain the training procedure, including the definition of training and test sets, as well as the choice of an optimization algorithm. Finally, we benchmark the resulting and other noise models, particularly the device noise model provided in qiskit.

4.1. Constructing the noise model

The noise model that we use later for training and benchmarking describes quantum computers similar to IBM's `ibmq_manila` device using a Falcon processor. It can be easily generalized to any other gate-based quantum device with few minor adaptions. The `ibmq_manila` machine has $N = 5$ qubits in a linear layout (see figure 2). It implements three single-qubit gates (X, $\sqrt{\mathsf{X}}$,R_z) and one two-qubit gate (CNOT) as native gates. We denote the native gate set by $\mathcal{G} = \{\mathsf{X},\ \sqrt{\mathsf{X}},\ \mathsf{R}_z,\ \mathsf{CNOT}\}$.

Our noise model can describe any device with N qubits in a linear layout. For other layouts, adaptions must be made to the possible multi-qubit interactions. The model combines all types of noise defined in section 2.1. At the beginning of each computation, the initial state is prepared as $\rho_0 = |0\cdots 0\rangle\langle 0\cdots 0|$. It is followed by state preparation error $\mathcal{S}$ with corresponding probabilities $p_{\mathrm{sp}}(q)$, yielding N model parameters.

Afterwards, gates are applied to the resulting (possibly erroneous) state. Each gate g is followed by

- crosstalk error $\mathcal{C}$ (for $g \in \{\mathsf{X},\ \sqrt{\mathsf{X}}\}$) with parameter $\phi_g(q)$, applied to the neighbour qubits,
- depolarizing error $\mathcal{D}$ with parameters $\lambda_g(q)$ for single-qubit gates and $\lambda_g(q_1, q_2)$ for the CNOT gate, applied to the gate qubit(s),
- and thermal relaxation and dephasing $\mathcal{T}$ with parameters $T_{1,2}(q)$, applied to the gate qubit(s).

After all gates and their errors have been applied, measurement error $\mathcal{M}$ affects all qubits with parameters $p_{0,1\to 1,0}(q)$. Finally, the qubits are measured in the computational basis. A basic example of our noise model on a quantum circuit containing only one X gate and one CNOT gate can be found in figure 3.

The total number of parameters of our noise model for a system of N is $11N - 1$, as shown in table 1. Note that we assume the CNOT gates to only be applied in one direction per qubit pair. Since there are three types on one-qubit gates and one two-qubit gate, there are $4N - 1$ model parameters corresponding to depolarization error.

4.2. Simulating the noise model

All noisy simulations using the above model are carried out by exactly computing the density matrix of the system and its change due to errors. As section 2.1 explains, all errors included in the model can be represented by quantum operations, which are linear maps of the density matrix. These linear maps depend on the respective parameters of the errors, e.g., bit-flip probabilities in the case of readout error. Pennylane [48] offers the possibility to implement the errors and simulate quantum circuits with our noise model.

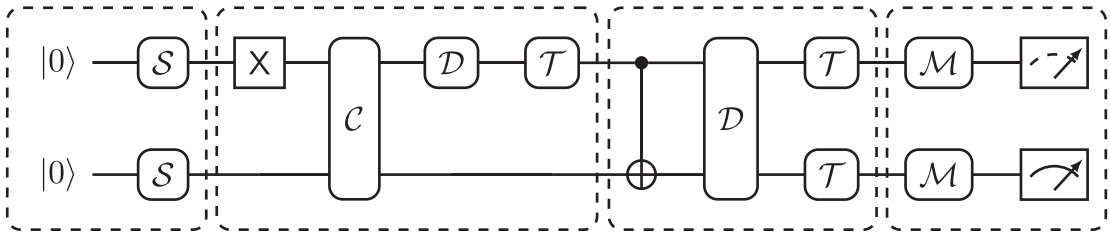


Figure 3. Example of a quantum circuit subject to our noise model. $\mathcal{S}$, $\mathcal{C}$, $\mathcal{D}$, $\mathcal{T}$, and $\mathcal{M}$ denote state preparation, crosstalk, thermal relaxation, depolarization, and measurement error, respectively. Note that the crosstalk error is depicted on both qubits to emphasize their interaction. The dashed boxes indicate what circuit operations are affected by which errors.

Table 1. Number of parameters corresponding to each type of error in the noise model, assuming a system of N qubits. The parameters are later optimized during model training. See figure 1 and (1)-(4) for more details.

symbol	error	parameters	number of parameters
$\mathcal{S}$	state preparation	$p_{\mathrm{sp}}(q)$	N
$\mathcal{D}$	depolarization	$\lambda_g(q)$	$4N-1$
$\mathcal{C}$	crosstalk	$\phi_g(q)$	$2N$
$\mathcal{T}$	thermal relaxation	$T_{1,2}(q)$	$2N$
$\mathcal{M}$	measurement	$p_{0\to1}(q)$, $p_{1\to0}(q)$	$2N$
total			$11N-1$

After initializing the density matrix, all gates and errors are applied, and the final density matrix is computed. From this final density matrix, we obtain one of the following two quantities: in training, we compute the outcome distribution of basis states, while for the benchmarks, we use the expectation value of the $\mathsf{Z}^{\otimes w}$ operator.

If we take the situation from figure 3 as an example, the initial density matrix is $\rho_0 = |00\rangle\langle 00|$. Afterwards, (1) is used to apply state preparation error with probability $p_{\mathrm{sp}}(0)$ to the first and with probability $p_{\mathrm{sp}}(1)$ to the second qubit. Next, the X gate acts on the first qubit, followed by crosstalk, depolarizing, and thermal relaxation error. These errors are computed using the corresponding equations from section 2.1 with parameters $\phi_{\mathsf{X}}(0)$, $\lambda_{\mathsf{X}}(0)$, and $T_{1,2}(0)$, respectively. The rest of the computation is done similarly.

4.3. Training the noise model

This section explains how the parameters of our noise model from section 4.1 can be optimized. The approach is inspired by machine learning in the sense that model predictions are repeatedly evaluated on a training data set, and parameters are adapted accordingly.

For the training, quantum circuits are first compiled to the native gate set $\mathcal{G}$. The compiled circuits are then both run on a quantum computer and simulated with the noise model. Afterwards, a loss function is defined that measures the deviation of model predictions from hardware outcomes of the quantum circuits. Details on the implementation can be found in appendix C.

Training set

The training set contains 100 quantum circuits for which the noise model predicts noisy outcomes. Since we are mainly interested in VQE quantum computing applications, these training circuits consist of alternating layers of single-qubit rotations and entanglement. They follow the `EfficientSU2` structure that is part of the qiskit library. The number of layers is denoted as d. Increasing the number of quantum circuits contained in the training set did not further improve the parameter optmization.

Each rotational layer applies an R_y gate followed by an R_z gate to each qubit. Their rotation angles are randomized. The entanglement layers consist of CNOT gates that linearly connect all qubits. For example, the resulting quantum circuit for $d = 3$ layers and $w = 2$ qubits is shown in figure 4.

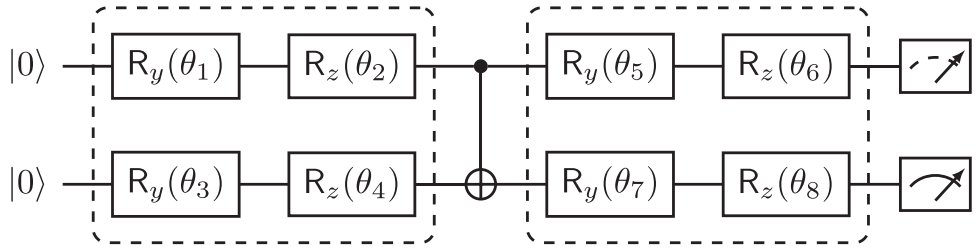


Figure 4. `EfficientSU2` circuit with three layers on two qubits. Each rotational layer consists of an R_y gate followed by an R_z gate on each qubit. Entanglement layers implement CNOT gates acting linearly on all qubit pairs.

Table 2. Comparison of noise channels included in different models. The brackets indicate readout error with only symmetric probabilities.

	readout	qiskit	[38]	trained
state preparation	X	X	✓	✓
depolarisation	X	✓	✓	✓
crosstalk	X	X	X	✓
thermal relaxation	X	✓	✓	✓
measurement	✓	✓	(✓)	✓

4.3.1. Loss function

The loss function compares the model predictions to the outcomes of the quantum circuits run on the device. The latter ones are measurement counts of computational basis states. Exact density matrix simulations allow to compute the probability distribution of these measurements based on the noise model. If one interprets the hardware counts as relative frequencies, the task is to compare two probability distributions.

Various metrics measure the distance between two distributions, e.g., Kullback-Leibler (KL) divergence [49]. Here, we use the Hellinger distance [50] between the probability distributions from the simulation and hardware run to define the loss on a single quantum circuit. If $P = (p_i)_{i\in\mathcal{I}}$ and $Q = (q_i)_{i\in\mathcal{I}}$ are two discrete probability distributions, then their Hellinger distance $H(P, Q)$ is defined as

$$H(P,\,Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i\in\mathcal{I}}(\sqrt{p_i} - \sqrt{q_i})^2}. \tag{5}$$

For a set of multiple circuits, as in training, we define the loss function as the arithmetic mean of Hellinger distances. Training the noise model with KullbackLeibler divergence leads to similar results.

4.3.2. Optimization algorithm

For optimizing the parameters of our noise model, we use the simultaneous perturbation stochastic approximation (SPSA) algorithm [15]. Since each evaluation of the loss function involves simulating 100 quantum circuits, the optimizer must should only a few evaluations to be cost-effective. The SPSA algorithm approximates gradients with only two evaluations of the loss function per iteration. Therefore, it is well suited for this task.

4.4. Other noise models

Besides the trained noise model from above, we benchmark two others. The first one only includes readout error, where the bit-flip probabilites are obtained from the device calibration by IBM. It serves as a basic example to explain our approach here and can be simulated exactly using density matrices.

The second other noise model is the `ibmq_manila` device noise model provided in qiskit. Since its calibration changes with time, we always use the corresponding snapshots of the model when comparing it to hardware data. Moreover, simulations with the device noise model are always done shot-wise, meaning that we cannot compute predictions exactly. We mitigate possible variance effects by using large shot numbers.

Table 2 summarizes all noise models described above. It also contains information about the model presented in [38], which is similar to our model. The main difference is that it only considers symmetric readout error and does not contain crosstalk.

Table 3. Execution times of hardware experiments for volumetric benchmark. All experiments were run on the `ibmq_manila device` in September 2022. The times are in UTC+2.

	$d = 1$	$d = 2$	$d = 3$	$d = 4$	$d = 5$
$w = 1$	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

4.5. Volumetric benchmarks

This section describes how the volumetric benchmarks of different noise models were conducted. It follows the procedure from section 3.1. For the benchmarks, the `ibmq_manila` device was used. Therefore, the native gate set is $\mathcal{G} = \{\mathsf{X}, \sqrt{\mathsf{X}}, \mathsf{R}_z, \mathsf{CNOT}\}$, and the maximal number of qubits for the benchmark is $w = 5$. Further implementation details can be found in appendix C.

1. **Test circuits.** Similarly to the training, we use `EfficientSU2` circuits for benchmarking. They have alternating layers of rotational and entanglement gates. All odd layers consist of an R_y gate followed by an R_z gate, starting with the first layer. In between, there are CNOT gates that linearly connect all qubits.

   For width w and depth d, the set $C(w, d)$ consists of 200 circuits with d layers acting on w qubits. The rotation angles are randomized for each circuit. Note that we use different circuits for training and benchmarking.

2. **Compilation rules.** Different types of optimization can be applied during the compilation of quantum circuits. With no optimization, every gate of the circuit is compiled into a representation by native gates. Otherwise, the number of gates in the resulting circuit is minimized to reduce the impact of quantum noise.

   The qiskit library offers several configurations for this process, which are applied by the `transpile` function and its `optimization_level` argument. We choose `optimization_level=2` for compilation.

3. **Model predictions.** Exact simulations based on density matrices are used here to predict noisy $\mathsf{Z}^{\otimes w}$ expectations values of the test circuits, see section 4.2 and appendix C for further information.

   If one wants to benchmark the device noise model from qiskit, only shot-wise simulations of the circuits are supported. Therefore, the outcomes are measurement counts of basis states, as in the case of hardware results. We use 8192 shots for each simulation, which is sufficient to reduce shot noise, i.e., statistical uncertainty due to finite shot numbers.

4. **Hardware results.** The hardware results of the test circuits are obtained with the `ibmq_manila` device. As for the simulations above, every circuit is run 8192 times to reduce shot noise.

   Since the jobs are placed in a queue, the experiments for different pairs (w,d) cannot be run simultaneously. We save a snapshot of the device noise model before each run to enable its fair evaluation later. Hence, comparing different noise models does not depend on the execution time. The running times for each experiment can be found in table 3.

5. **Single circuit evaluation.** To compare the model predictions to the hardware results for a single quantum circuit $c \in C(w, d)$, we evaluate the expectation values of the $\mathsf{Z}^{\otimes w}$ operator and compute their absolute difference

$$d(c) = |\langle \mathsf{Z}^{\otimes w}_{\text{model}}(c)\rangle - \langle \mathsf{Z}^{\otimes w}_{\text{hardware}}(c)\rangle|. \tag{6}$$

6. **Overall evaluation.** We compute the arithmetic mean of all single-circuit results for the overall evaluation:

$$L = \frac{1}{n} \sum_{c \in C(w,d)} d(c),$$

   where n is the number of circuits in $C(w, d)$.

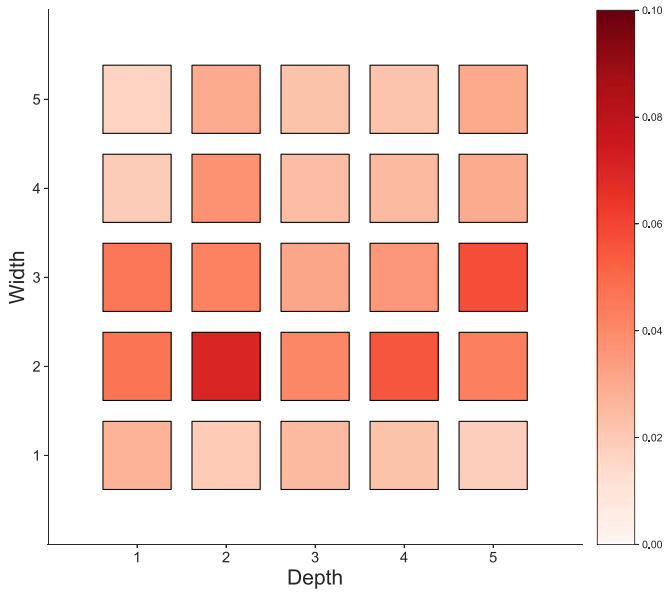


Figure 5. Volumetric benchmark results for readout noise model with calibrated bit-flip probabilities from section 4.4. The colours represent the average absolute deviation of the predicted $Z^{\otimes w}$ expectation value from the hardware data. Darker squares indicate larger deviations and worse model accuracy.

4.6. Confidence intervals

To estimate the statistical significance of our results, we perform bootstrapping and compute confidence intervals based on the resulting bootstrap distributions. The bootstrapping procedure is as follows. Since model predictions are based on exact density matrix calculations, the only uncertainty for our benchmarks stems from the hardware results $\langle Z^{\otimes w}_{\text{hardware}} \rangle$. The results of the experiments consist of 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 one of the 8192 outcomes from the hardware. We repeat this procedure $b = 10^5$ 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 computed using propagation of uncertainty:

$$u = \frac{1}{200}\sqrt{\sum_c u_c^2}. \tag{7}$$

5. Results

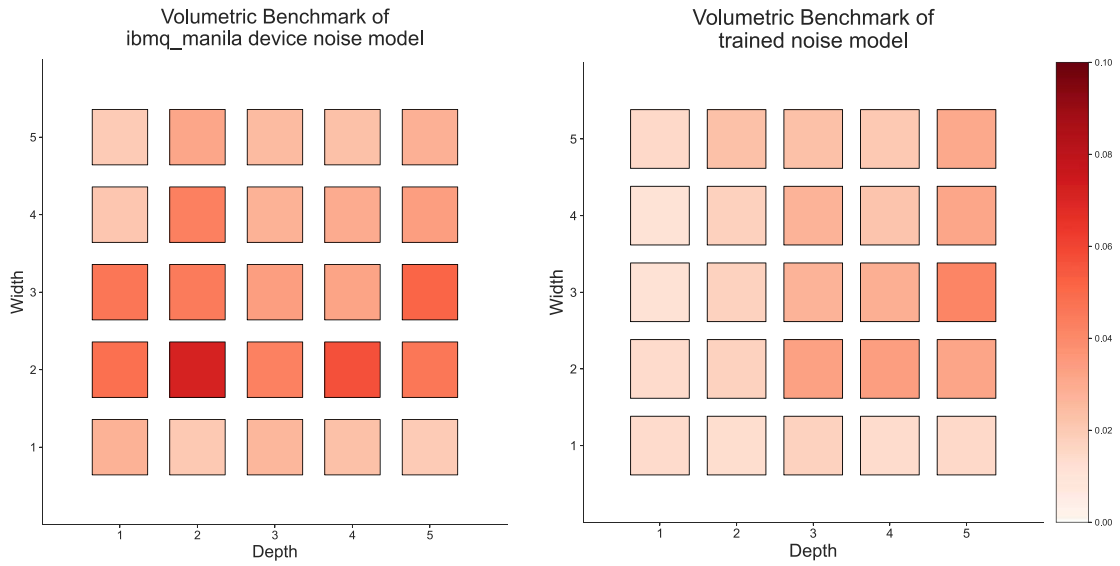
This section presents the results of volumetric benchmarks for the noise models from above. Details on the benchmark process and noise models are given in section 4. Recall that the volumetric benchmark compares $Z^{\otimes w}$ expectation values of noisy simulation and hardware experiment for different widths w and depths d. These expectation values are bounded by the interval $[-1, 1]$, restricting the absolute difference between two such values to a maximum of 2. The result of each configuration (w,d) is represented by a different square in the figure. The overall style of presentation is inspired by [14].

The colour of a square indicates the average absolute deviation between noise model prediction and hardware data. Darker squares indicate a larger deviation, while white squares indicate good agreement. On the right side of the plot, one can find a legend explaining how the colours translate to numeric values. This legend is valid for all three plots, so the benchmark results for all noise models can be directly compared.

Consider the readout model and its benchmark results in figure 5 as an instructive example. The figure shows that the noise model predicts the hardware behaviour well for $w = 1$, i.e., for a single qubit. For larger qubit numbers, the deviations between model predictions and hardware data increase. For example, one finds an average absolute error of $Z^{\otimes w}$ of almost 0.1 for $w = w, d = 2$.

Figure 6 shows the volumetric benchmark results for the device model and our trained model, where the former can be found in figure 6(a) and the latter in figure 6(b).

Figure 7 shows the confidence intervals of each benchmark based on the procedure from section 4.6. The blue, striped bars represent our trained noise model, while the red bars show the results of the `ibmq_manila` device noise model.



(a) `ibmq_manila` device noise model. (b) Trained noise model.

Figure 6. Volumetric benchmark results for the `ibmq_manila` device noise model and the trained noise model.

5.1. Discussion

The three noise models perform very differently in the volumetric benchmark. Their results improve with model complexity, meaning that our trained model achieves the best results, followed by the qiskit device noise model. In the following, we discuss the volumetric benchmarks in more detail.

Figure 6(a) shows the benchmark results for qiskits device noise model. Except for some negative outliers, such as for $w = 2$ and $d \in \{2, 4\}$, the accuracy of model predictions remains stable for different configurations (w,d). The results are similar to the readout noise model from figure 5, showing that the readout error is the dominant source of noise in the qiskit device model. The accuracy of these models strongly depends on the calibration procedure. If parameters are calibrated incorrectly, the deviation between noisy simulation and hardware experiments increases. This is one possible reason for the outliers mentioned above. The device noise model can provide an easily accessible way to simulate quantum circuits with a certain confidence. However, its accuracy is not optimal for realistic simulations. This could change in later versions of qiskit with more error types included.

As shown in figure 6(b), our noise model with optimized parameters achieves good overall benchmark results. Its worst performance is an average deviation in $\mathbf{Z}^{\otimes w}$ expectation value of 0.043 (compared to 0.067 of the previous model). The model works particularly well for shallower quantum circuits with up to three layers. For deeper circuits, we observe a slight decrease in its accuracy, closing the gap between the two models. Since the qiskit noise model contains a subset of the noise channels from the trained model, the latter should achieve equal or better results everytime, given optimal model parameters. The primary obstacle for its performance is the training procedure, which becomes increasingly more difficult for larger parameter numbers.

The good performance of our model is also supported by figure 7. For all shallow quantum circuits with $d \leqslant 2$, as well as for small qubit numbers with $w \leqslant 2$, it shows a significant improvement compared to the device noise model. For all other configurations, our model either performs better or equally well within the statistical confidence.

5.2. Limitations and threats to validity

In the following, we discuss limitations and threats to validity concerning both contributions from section 1.1.

The framework

There are two main limitations of our benchmarking framework for noise models. Firstly, noisy simulations of quantum circuits become computationally intractable for large qubit numbers. The density matrix of N qubits has dimension $2^N \times 2^N$, making exact computations exponentially expensive. While the statevector of the same system has only 2^N entries, enabling stochastic methods such as in [51] on more qubits than full density matrix computations, the scaling remains exponential. This limitation is not specific to our approach but ubiquitous in the field of quantum computing. Since our aim is to provide noise model benchmarks for the NISQ era, small

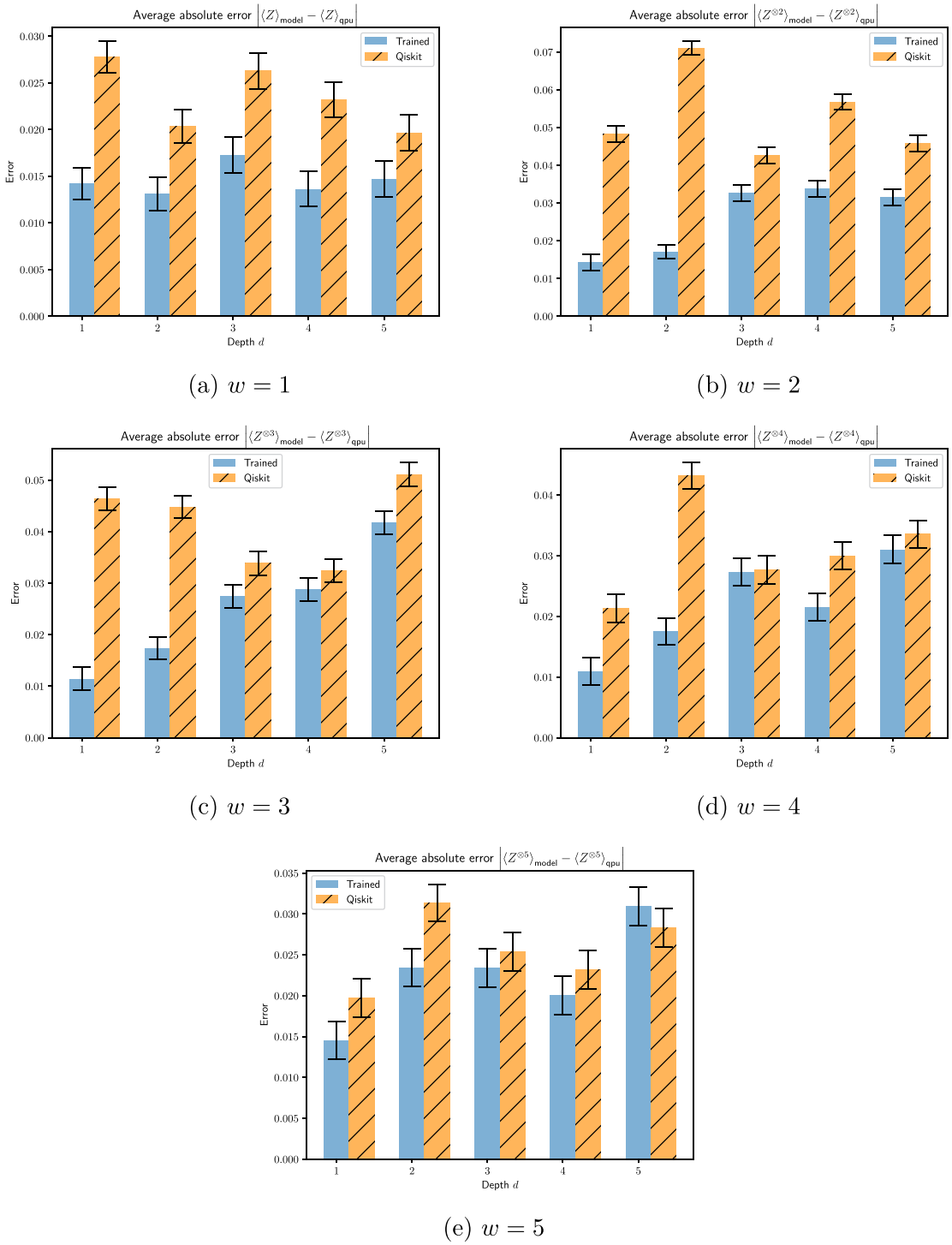


Figure 7. Confidence intervals for average absolute error of $Z^{\otimes w}$ expectation value. The blue, striped bars represent the results for our trained noise model, while the orange bars show the results for the qiskit device noise model. The x-axes of the plots indicate the depth d of the quantum circuits.

systems are the primary focus of our approach. Moreover, simulations could be simplified under certain locality assumptions on the noise and with restricted qubit connectivity.

Secondly, the volumetric framework does not automatically ensure quality in terms of the criteria from section 2.2. This quality depends on user choices for test circuits, evaluation metrics, and other specifications. However, we explained in detail in section 3.2 how these choices can be made to improve the benchmark quality for each individual criterion.

The benchmarks

There are three threats to validity that we identify for our benchmarks. These threats potentially affect the quality criteria relevance, reproducibility, and fairness.

Firstly, hardware results and predictions from the device noise model are obtained using a finite number of shots. Thus, the benchmark results are subject to statistical noise and repeating the benchmarks can yield different outcomes. We mitigate this threat with a large number of shots and quantum circuits.

Secondly, the hardware experiments were conducted at different times. Since the noise level in a quantum computer is not constant, the ideal noise model is not always the same. Therefore, comparing the benchmark results of a noise model for one configuration (w,d) to another is not necessarily meaningful. Instead, one should compare the results of different noise models for fixed (w,d).

Thirdly, the quantum circuits used for the benchmarks are specific to variational algorithms. Our results are not necessarily generalizable to other applications of quantum computing that use different types of circuits, for example for factorization or search algorithms.

6. Conclusion

6.1. Summary

This paper presents a novel approach to evaluate the accuracy of quantum computing noise models. The approach is based on volumetric benchmarks that compare model predictions to the behaviour of a quantum device for sets of quantum circuits of different sizes. If a noise model performs well in these volumetric benchmarks, it can be used for noisy simulations, reducing the need for quantum hardware. Possibilities to improve the benchmark quality in terms of established quality criteria are also discussed.

We conducted volumetric benchmarks for three noise models using the `ibmq_manila` quantum computer. The first noise model only considers readout error with calibrated probabilities. The second is the device noise model for the `ibmq_manila` hardware from the qiskit library. We construct a third model with trainable parameters that we optimize using a set of training circuits. It contains SPAM error, depolarizing error, thermal relaxation and dephasing, and a simple form of crosstalk error. More types of noise can easily be added to the model.

While the readout noise model performed poorly for more than a single qubit, the device and the trained noise model achieved better results for larger system sizes. The predictions of the former still showed larger deviations from hardware data for several configurations of width w and depth d. In particular, the accuracy for the configurations $w = 2,3$ is decreased. The trained noise model performs significantly better for small qubit numbers ($w \leqslant 2$). Except for the configurations (3, 4), (3, 5), (4, 3), (4, 5), and for $w = 5$, $d \geqslant 3$, where no statistically significant statement can be made, it shows improved results compared to the device noise model. Overall, its accuracy is stable for most configurations. Only for deep quantum circuits do we find a slight decrease. The reason could be a more demanding training environment. Overall, our noise model and approach to training its parameters show promising results in these first volumetric benchmarks.

6.2. Future work

The noise model constructed in this paper includes a simple form of coherent crosstalk error. As explained in section 2.1 and in more detail in [45], crosstalk can be very versatile and is not necessarily coherent. Therefore, future research should construct noise models with more complex descriptions of crosstalk to further improve our understanding of quantum noise. Other types of noise should also be considered.

Furthermore, future research should conduct more extensive volumetric benchmarks. This includes quantum hardware with more qubits and quantum circuits from a larger variety of applications.

Moreover, new evaluation criteria for volumetric benchmarks should be investigated to explore other quantum computing applications. While the expectation value of observables is of interest for VQE, different variables are more significant for other algorithms such as Grover [52].

Finally, the training method presented in this paper can be improved for better parameter optimization of noise models.

Acknowledgments

We acknowledge the support by DASHH (Data Science in Hamburg—HELMHOLTZ Graduate School for the Structure of Matter) with the Grant-No. HIDSS-0002. This research was funded by Deutsches Elektronen-Synchrotron DESY, a member of the Helmholtz Association (HGF). This work is supported with funds from the Ministry of Science, Research and Culture of the State of Brandenburg within the Centre for Quantum Technologies and Applications (CQTA).

Data availability statement

The data cannot be made publicly available upon publication because they are not available in a format that is sufficiently accessible or reusable by other researchers. The data that support the findings of this study are available upon reasonable request from the authors.

Appendix A. Density operators

A quantum system can be described by a Hilbert space $\mathcal{H}$ and a bounded, self-adjoint operator ρ: $\mathcal{H} \to \mathcal{H}$ called the *density operator* (or *density matrix*). It is defined to satisfy the following properties:

- ρ is positive.
- ρ is trace class with $\mathsf{tr}(\rho) = 1$.

The evolution of such a system with a unitary operator U can be expressed as a mapping

$$\rho \mapsto U\rho U^{\dagger}.$$

Measurements are described by a set $\{M_i\}_{i\in\mathcal{I}}$ of measurement operators such that

$$\sum_{i\in\mathcal{I}} M_i^{\dagger} M_i = \mathsf{I}.$$

For any observable A: $\mathcal{H} \to \mathcal{H}$, its quantum mechanical *expectation value* is given by

$$\langle A\rangle_{\rho} = \mathsf{tr}(A\rho).$$

Moreover, if the system is in state ρ_i with probability p_i, then its density operator is

$$\rho = \sum_i p_i \rho_i.$$

For quantum computing, density operators can be applied as follows. As qubits are two-dimensional quantum systems, they are described by a two-dimensional Hilbert space $\mathcal{H} \simeq \mathbb{C}^2$ with the so-called *computational basis* $\{|0\rangle, |1\rangle\}$, where

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

A qubit state can then be expressed in terms of a 2×2 density matrix ρ with $\mathsf{tr}(\rho) = 1$. In quantum computing, qubits are prepared in an initial state, manipulated by unitary gates, and finally measured in the computational basis. Typically, the initial state is $|0\rangle$. The corresponding density matrix is

$$\rho_0 = |0\rangle\langle 0| = \begin{pmatrix}1 & 0\\0 & 0\end{pmatrix}.$$

For a composite system of N qubits, the Hilbert space and initial state are $\mathcal{H} = \mathbb{C}^{2N}$ and $\rho_0 = |0\cdots 0\rangle\langle 0\cdots|$, respectively.

Appendix B. Quantum operations

The term *quantum noise* labels all processes not part of the intended quantum circuit consisting of state preparation, gate operations, and measurements. Quantum operations are a powerful tool for expressing these processes in terms of density operators. Roughly speaking, a quantum operation $\mathcal{E}$ maps the density operator ρ of a quantum system with Hilbert space $\mathcal{H}$ to a density operator ρ' of $\mathcal{H}'$: $\rho' = \mathcal{E}(\rho)$.

Mathematically, a quantum operation $\mathcal{E}$ from a Hilbert space $\mathcal{H}$ to a Hilbert space $\mathcal{H}'$ is a linear map between their sets of positive trace class operators such that

- if ρ is a density operator, then $\mathsf{tr}(\mathcal{E}(\rho)) \leqslant 1$
- $\mathcal{E}$ is completely positive.

We do not discuss this definition in more detail, instead we refer to the literature for further reading [39].

Kraus' theorem [44] gives a helpful characterization of quantum operations. It states that a linear map $\mathcal{E}$ between the spaces mentioned above is a quantum operation if and only if there is a set of linear operators $\{O_i\colon \mathcal{H} \to \mathcal{H}'\}$ such that

$$\mathcal{E}(\rho) = \sum_i O_i \rho O_i^\dagger$$

with $\sum_i O_i^\dagger O_i \leqslant \mathsf{I}$. An important example of a quantum operation for quantum computing is the *depolarizing error*

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

where d is the dimension of the system, i.e. $D = 2^N$ for N qubits. Denoting the Pauli matrices by X, Y and Z, the depolarizing error on a single qubits takes the following form in terms of Kraus operators:

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

Appendix C. Implementation details

This section explains the implementation of the training and benchmarking of noise models in more detail. It contains three parts that discuss simulating quantum circuits, running experiments on quantum hardware, and optimizing parameters.

C.1. Hardware experiments

For 25 possible configurations of (w,d), 100 training circuits and 200 benchmark circuits were run on the `ibmq_manila` quantum computer. The circuits of a pair (w,d) were first compiled into native gates using the `transpile` method with the argument `optimization_level=2` from the qiskit software library (qiskit version 0.38.0). Afterwards, they were sent to the device as one job and executed consecutively. The running times of the experiments can be found in table 3. Snapshots of the qiskit device noise model are saved at every run.

C.2. Noisy simulations

Similar to the hardware experiments, all circuits are compiled into native gates. The compiled circuits are then simulated with different noise models. We use two software packages for the Python programming languages for these simulations.

Simulations of the ibmq_manila device noise model are implemented using qiskit and its `AerSimulator` device with the noise model saved at the corresponding hardware run. We always use 8192 shots.

Simulations of our noise model are implemented using Pennylane using the `default.mixed` device. This device allows for exact computations of the density matrix and, therefore, for exact predictions of outcome probabilities or expectation values. During the training of the noise model, the outcome distribution is computed using `probs` measurements. The $\mathsf{Z}^{\otimes w}$ expectation value is calculated with the `expval` measurement for the volumetric benchmarks.

C.3. Parameter training

The noise model parameters are trained using the SPSA algorithm. At every iteration of the optimization process, the loss function from section 4.3 is evaluated on 100 `EfficientSU2` quantum circuits with randomized parameters. The Hellinger distance is computed by comparing the probability distribution of the noisy simulation to the counts of the hardware run. The latter are interpreted as distribution via their relative frequencies.

The optimizer trains for 500 epochs with the hyperparameter c set to $c = 0.005$. The a hyperparameter varies between $a = 0.005$ and $a = 0.08$, depending on w and d. Moreover, we use $\alpha = 0.602$ and $\gamma = 0.101$, as recommended in [53].

ORCID iDs

Tom Weber https://orcid.org/0000-0002-7560-4963

References

[1] Preskill J 2018 *Quantum* **2** 1–20
[2] Endo S, Benjamin S C and Li Y 2018 *Phys. Rev.* X **8** 1–20
[3] Peruzzo A, McClean J, Shadbolt P, Yung M H, Zhou X Q, Love P J, Aspuru-Guzik A and O'Brien J L 2014 *Nat. Commun.* **5** 1–10
[4] Bravyi S, Sheldon S, Kandala A, Mckay D C and Gambetta J M 2021 *Phys. Rev.* A **103** 42605
[5] Funcke L, Hartung T, Jansen K, Kühn S, Stornati P and Wang X 2022 *Phys. Rev.* A **105** 062404

[6] Kandala A, Temme K, Córcoles A D, Mezzacapo A, Chow J M and Gambetta J M 2019 *Nature* **567** 491–5
[7] Kwon H and Bae J 2021 *IEEE Trans. Comput.* **70** 1401–11
[8] Murali P, McKay D C, Martonosi M and Javadi-Abhari A 2020 Software Mitigation of Crosstalk on Noisy Intermediate-Scale Quantum Computers *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture* 279–90
[9] Su D, Israel R, Sharma K, Qi H, Dhand I and Brádler K 2021 *Quantum* **5** 452
[10] Sun J, Yuan X, Tsunoda T, Vedral V, Benjamin S C and Endo S 2021 *Physical Review Applied* **15** 34026
[11] Temme K, Bravyi S and Gambetta J M 2017 *Phys. Rev. Lett.* **119** 1–15
[12] Vovrosh J, Khosla K E, Greenaway S, Self C, Kim M S and Knolle J 2021 *Phys. Rev.*E **104** 035309
[13] Takagi R, Endo S, Minagawa S and Gu M 2022 *npj Quantum Information* **8** 1–11
[14] Blume-Kohout R and Young K C 2020 *Quantum* **4** 362
[15] Spall J C 1992 *IEEE Trans. Autom. Control* **37** 332–41
[16] ANIS M S *et al* (2021) *Qiskit: An Open-Source Framework for Quantum Computing* (https://doi.org/10.5281/zenodo.2573505)
[17] Greenbaum D 2015 *Introduction to Quantum Gate Set Tomography* (https://doi.org/10.48550/arXiv.1509.02921)
[18] Leibfried D, Meekhof D M, King B E, Monroe C, Itano W M and Wineland D J 1996 *Phys. Rev. Lett.* **77** 4281–5
[19] Altepeter J B, Branning D, Jeffrey E, Wei T C, Kwiat P G, Thew R T, O'Brien J L, Nielsen M A and White A G 2003 *Phys. Rev. Lett.* **90** 193601
[20] Chuang I L and Nielsen M A 1997 *J. Mod. Opt.* **44** 2455–67
[21] D'Ariano G M, Laurentis M D, Paris M G A, Porzio A and Solimeno S 2002 *J. Opt. B: Quantum Semiclassical Opt.* **4** S127
[22] Mohseni M, Rezakhani A T and Lidar D A 2008 *Phys. Rev.* A **77** 032322
[23] Poyatos J F, Cirac J I and Zoller P 1997 *Phys. Rev. Lett.* **78** 390–3
[24] Shabani A, Kosut R L, Mohseni M, Rabitz H, Broome M A, Almeida M P, Fedrizzi A and White A G 2011 *Phys. Rev. Lett.* **106** 100401
[25] Blume-Kohout R, Gamble J K, Nielsen E, Mizrahi J, Sterk J D and Maunz P 2013 arXiv:1310.4492
[26] Merkel S T, Gambetta J M, Smolin J A, Poletto S, Córcoles A D, Johnson B R, Ryan C A and Steffen M 2013 *Phys. Rev.* A **87** 062119
[27] Nielsen E, Gamble J K, Rudinger K, Scholten T, Young K and Blume-Kohout R 2021 *Quantum* **5** 557
[28] Blume-Kohout R, da Silva M P, Nielsen E, Proctor T, Rudinger K, Sarovar M and Young K 2022 *PRX Quantum* **3** 020335
[29] Emerson J, Alicki R and Życzkowski K 2005 *J. Opt. B: Quantum Semiclassical Opt.* **7** S347–52
[30] Emerson J, Silva M, Moussa O, Ryan C, Laforest M, Baugh J, Cory D G and Laflamme R 2007 *Science* **317** 1893–6
[31] Knill E, Leibfried D, Reichle R, Britton J, Blakestad R B, Jost J D, Langer C, Ozeri R, Seidelin S and Wineland D J 2008 *Phys. Rev.* A **77** 012307
[32] Erhard A, Wallman J J, Postler L, Meth M, Stricker R, Martinez E A, Schindler P, Monz T, Emerson J and Blatt R 2019 *Nat. Commun.* **10** 5347
[33] Cross A W, Bishop L S, Sheldon S, Nation P D and Gambetta J M 2019 *Phys. Rev.* A **100** 032328
[34] Proctor T, Rudinger K, Young K, Nielsen E and Blume-Kohout R 2022 *Nat. Phys.* **18** 75–9
[35] Cirstoiu C, Dilkes S, Mills D, Sivarajah S and Duncan R 2023 *Quantum* **7** 1059
[36] Blume-Kohout R, Rudinger K, Nielsen E, Proctor T and Young K 2020 arXiv:2012.12231
[37] Harper R, Flammia S T and Wallman J J 2020 *Nat. Phys.* **16** 1184–8
[38] Georgopoulos K, Emary C and Zuliani P 2021 *Phys. Rev.* A **104** 062432
[39] Nielsen M A and Chuang I L 2010 *Quantum Computation and Quantum Information* 10th edn (Cambridge University Press) (https://doi.org/10.1017/CBO9780511976667)
[40] Tannu S S and Qureshi M K 2019 Mitigating Measurement Errors in Quantum Computers by Exploiting State-Dependent Bias *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture* 279–90
[41] Geller M R and Sun M 2021 *Quantum Science and Technology* **6** 025009
[42] Sanders Y R, Wallman J J and Sanders B C 2015 *New J. Phys.* **18** 012002
[43] Zurek W H 1991 *Phys. Today* **44** 36–44
[44] Choi M D 1975 *Linear Algebr. Appl.* **10** 285–90
[45] Sarovar M, Proctor T, Rudinger K, Young K, Nielsen E and Blume-Kohout R 2020 *Quantum* **4** 321
[46] Kistowski J, Arnold J A, Huppler K, Lange K D, Henning J L and Cao P 2015 *How to build a benchmark Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering* 333–6
[47] Funcke L, Hartung T, Jansen K, Kühn S and Stornati P 2021 *Quantum* **5** 422
[48] Bergholm V *et al* 2018 *Pennylane: Automatic Differentiation of Hybrid Quantum-Classical Computations* (https://doi.org/10.48550/arXiv.1811.04968)
[49] Kullback S and Leibler R A 1951 *The Annals of Mathematical Statistics* **22** 79–86
[50] Hellinger E 1909 *Journal für die reine und angewandte Mathematik* **1909** 210–71
[51] Bassi A and Deckert D A 2008 *Physical Review A* **77** 032323
[52] Grover L K 1996 A fast quantum mechanical algorithm for database search *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* 212–9
[53] Spall J 1998 *IEEE Trans. Aerosp. Electron. Syst.* **34** 817–23