Benchmarking Quantum Computers and the Impact of Quantum Noise

SALONIK RESCH and ULYA R. KARPUZCU, University of Minnesota

Benchmarking is how the performance of a computing system is determined. Surprisingly, even for classical computers this is not a straightforward process. One must choose the appropriate benchmark and metrics to extract meaningful results. Different benchmarks test the system in different ways, and each individual metric may or may not be of interest. Choosing the appropriate approach is tricky. The situation is even more open ended for quantum computers, where there is a wider range of hardware, fewer established guidelines, and additional complicating factors. Notably, quantum noise significantly impacts performance and is difficult to model accurately. Here, we discuss benchmarking of quantum computers from a computer architecture perspective and provide numerical simulations highlighting challenges that suggest caution.

CCS Concepts: • **Computer systems organization → Quantum computing;**

Additional Key Words and Phrases: Benchmarking quantum computers, quantum noise

ACM Reference format:
Salonik Resch and Ulya R. Karpuzcu. 2021. Benchmarking Quantum Computers and the Impact of Quantum Noise. *ACM Comput. Surv.* 54, 7, Article 142 (June 2021), 35 pages.
https://doi.org/10.1145/3464420

1 INTRODUCTION

There are many ways to measure the performance of a computer.[1] Common ways have been measuring **operations per second (OPS)** or **floating-point operations per second (FLOPS)**. These are intuitive and easy to understand; however, they are generally poor metrics. The problem is that hardware could be designed to have a very high OPS/FLOPS but could perform poorly on real-world applications, which do not consist of monolithic blocks of arithmetic operations. A way to improve upon this is to measure the progress of a program rather than the number of operations it performs. For example, the HINT benchmark measures quality improvements per second, which measures the numerical accuracy improvement of the output in a given time [68]. While this can be insightful, again the main concern is that this does not accurately represent the real-world programs that will be run on the hardware.

[1]For clarification, we note benchmarks are operations that the system is asked to perform (programs) and metrics are measurable characteristics of the system when performing the benchmark.

Supported in part by NSF CCF-1553042.
Authors' addresses: S. Resch and U. R. Karpuzcu, University of Minnesota, 200 Union St SE, Minneapolis, Minnesota, 55455; emails: {resc0059, ukarpuzc}@umn.edu.

0360-0300/2021/06-ART142 $15.00
https://doi.org/10.1145/3464420

Generally, the best metric is the wall-time required to complete a program [91] if the program is representative of real-world applications. This concept has led to the creation of benchmarks that are samples of larger, industrially useful applications. SpecCPU [141] and Parsec [15] are popular suites in this vein. While this is a clear improvement, it is not without its issues. For one, the reduced size of the programs introduces estimation error on the performance. There are other, less obvious complications. For example, academic work commonly reports performance on Parsec for system evaluation. Now, the human made choices of which benchmarks to include in Parsec determine what the academic community considers to be important. This makes these choices critical, because if the selection is not representative, then these results can be misleading. Even further, having established benchmarks would allow for a hardware designer to "cheat" by making a system particularly good on only the specific applications.

The takeaway is that benchmarking is possible and useful, yet is tricky and can be misleading. It is difficult to create useful benchmarks, and it may be impossible to create universal ones. This same construct applies to quantum computing, except it is much more intricate. There are a number of complicating factors:

(1) Quantum hardware is more diverse than classical hardware;
(2) Quantum hardware is less developed, most systems have only a few qubits and cannot perform useful applications;
(3) Quantum algorithms are still being developed and it is unknown what applications will be the most useful;
(4) Quantum noise is not well understood and difficult to simulate, making characterization particularly challenging.

The dissimilarity of quantum hardware makes it hard to compare them to each other. This is not as much of an issue for classical hardware. While there has been a trend toward more diversified and specialized hardware in recent years, such as application specific integrated circuits, there is a general framework and almost all hardware is silicon CMOS based. This makes metrics, benchmarks, and general intuition portable across different devices. Currently, there are many different hardware approaches competing in quantum computing. Each is based on a different physical system with entirely different dynamics. For example, quantum computing can be performed in superconducting circuits, ions isolated in a vacuum, or in atoms embedded in silicon. These systems look very different from each other and each have unique advantages and deficiencies. Is it fair to compare them directly?

As quantum computing is early on in its development, there are only small-to-medium sized quantum computers in existence. Most systems are not capable of performing useful programs. This makes it difficult to create benchmarks for these systems that are representative of future real-world applications. Scaling to larger sizes is particularly difficult for quantum computers, hence benchmarks that can be run on these smaller systems are less likely to accurately represent the performance of scaled-up versions. This is where one would normally turn to simulation. Unfortunately, as the states in quantum computers are highly complex, they are not able to be efficiently simulated by classical computers. Hence, benchmarks must be tied to a physical experiment.

On a more fundamental level, it is even unsure what quantum applications will be useful. As the field of quantum computing is largely unexplored, and not well understood, it is believed that many of its advantages and potential are currently unknown. Exploration of quantum potential is not well captured by benchmarking [20].

Quantum computing faces many hardware challenges. Information is easily lost due to quantum noise, which causes decoherence of quantum states. The physical devices need near absolute isolation from the environment, making the systems large and difficult to scale. Due to this fragility,

benchmarking begins much lower in the system stack. Benchmarks even for 1-bit operations have been developed [40, 43, 84]. Even at this level, performance has been difficult to quantify. Accurately modeling quantum noise and determining the robustness of quantum operations has become the subject of much research [44, 84, 119, 159]. Noise can affect quantum programs differently, depending on their length and structure. Hence, noise is a significant complicating factor.

Thus, quantum computing inherits all the benchmarking complexity of classical computing, but introduces many additional complications. This makes it quite unclear what the best way is to evaluate a quantum system. In fact, the authors of Reference [20] argue that it is too early to develop a standard approach. They warn that quantum research is currently exploratory in nature and that benchmarks are inappropriate for this kind of work. In fact, it could even be detrimental due to the possibility of misguiding research efforts.

In this article we provide a quantitative comparison for different quantum strategies from a benchmarking perspective, in the presence of quantum noise, to pinpoint pitfalls and fallacies. We start with basics in Section 2. We discuss quantum noise and noise models in Section 3. A brief introduction to quantum metrics is supplied in Section 4. Benchmarking for single- and two-qubit systems is covered in Section 5, for near-term noisy computing systems in Section 6, and for fault-tolerant computing systems in Section 7. Sections 8 and 9 detail the simulation results. Finally, we conclude the article in Section 10.

2 QUANTUM PRIMER

In this section, we introduce background and context for quantum computing. Necessarily brief, this clearly cannot do it justice. Quantum mechanics is highly complex and defies intuition. To quote Richard Feynman, "If you think you understand quantum mechanics, you don't understand quantum mechanics." This seems even more applicable if one views quantum mechanics from the perspective of computer architecture [75]. But in an attempt to "understand quantum mechanics," we will attempt to cover key concepts. We recommend [64] as an introduction to quantum mechanics and [94] as an introduction to quantum computing.

Quantum mechanics describes the nature of the physical world. High temperatures and large sizes causes quantum mechanical effects to become less noticeable, and classical physics acts as a good approximation. But when one creates a system that is very small or very cold, only quantum mechanics can accurately describe the system and how it evolves in time. Under these conditions, states are noticeably quantized (take on discrete values), such as the discrete possible energy levels of electrons around the nucleus of atoms. We can assign logical values to these distinct states, which are then called *qudits*. Transitions between these states correspond to quantum logical operations. Qudits can have many possible values, for example there are many possible non-degenerate energy levels for an electron. However, it is often convenient to use only two of the possible states, such as the ground and first excited states, as these become analogous to classical bits and are less susceptible to noise [109]. These two-level qudits are called *qubits*. Qubits can be in both of their states simultaneously (superposition) and multiple qubits can have their states intertwined (entanglement). Hence, there is not only information in each qubit but *between* each qubit. As a direct consequence, quantum states can store an amount of information that is exponential in the number of qubits. This enables extreme compute capabilities if one is able to create a complex quantum state and reliably transform it in a meaningful way. Unfortunately, this is a difficult task. Pure[2] quantum states are extremely fragile and need near perfect isolation from the environment to exist. At the same time, we need to be able to interact with the quantum state to transform it.

[2]Pure states are quantum states that can be completely specified by state vectors.

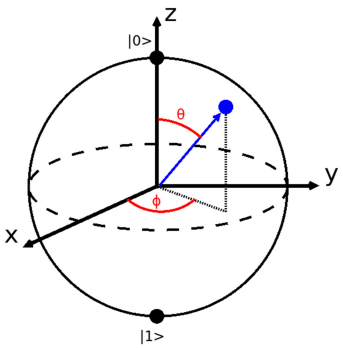


Fig. 1. Bloch Sphere representation of a single qubit.

Large-scale quantum computing, despite the fragility of quantum states, remains a possibility due to **quantum error correction (QEC)**. By encoding quantum information for a single qubit using multiple qubits, the quantum state can be restored if only a subset of the qubits become corrupted. Encoded qubits are called *logical* qubits, which are composed of multiple *physical* qubits. QEC is a rich field [60], and there is much work devoted to studying how QEC works under different error models [26, 63, 69, 71, 100]. However, modern quantum computers do not yet have sufficient numbers of qubits or required qubit quality to practically implement QEC. Hence, modern quantum applications operate on physical qubits and try to make use of limited resources. Therefore, it is of interest not only how quantum error affects QEC but also how it affects algorithms running without QEC.

If a quantum state is completely isolated, then it is called a *pure state*. Quantum pure states can be represented by *kets*, which are column vectors of complex numbers. The elements of the kets are called the *amplitudes*. For example, a single qubit can be represented by a ket of length 2,

$$\alpha\,|0\rangle + \beta\,|1\rangle = \begin{bmatrix}\alpha\\ \beta\end{bmatrix}, \tag{1}$$

where α is the amplitude associated with state $|0\rangle$ and β with state $|1\rangle$. The amplitudes determine probabilities when performing measurements, $|\alpha|^2$ is the probability of measuring this single qubit in the state $|0\rangle$ and $|\beta|^2$ is the probability of measuring the qubit in the $|1\rangle$ state. The probabilities must sum to 1 for pure states. A single qubit can be visualized as a vector from the origin to a point on the Bloch Sphere, shown in Figure 1, where a pair of angles, θ and ϕ, can be used to specify the state of the qubit, related to the amplitudes by the equations

$$\alpha = cos(\theta/2), \tag{2}$$

$$\beta = e^{i\phi} sin(\theta/2). \tag{3}$$

Operations on qubits are called *gates*, which are represented by matrices. Common gates are the Pauli **I** (identity), **X** (NOT), **Z** (phase-flip), and **Y** (NOT and phase flip) gates, shown in Equation (4),

$$\mathbf{I} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} \qquad \mathbf{Y} = \begin{bmatrix}0 & -i\\ i & 0\end{bmatrix} \qquad \mathbf{Z} = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}. \tag{4}$$

Single qubit gates represent rotations on the Bloch sphere. Performing a gate is logically equivalent to multiplying the ket column vector by the matrix of the gate. The **I** gate leaves the qubit unmodified. The **X** gate rotates the qubit by π around the X axis, which flips the amplitudes for the $|0\rangle$ and $|1\rangle$ states. Similarly, the **Z** gate rotates the qubit around the Z axis and the **Y** gate rotates

the qubit around the Y axis. Other common gates include the Hadamard (**H**) gate and phase gates **S** and **T**, shown in Equation (5),

$$\mathbf{H} = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \qquad \mathbf{S} = \begin{bmatrix}1 & 0\\ 0 & i\end{bmatrix} \qquad \mathbf{T} = \begin{bmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{bmatrix}, \tag{5}$$

The **S** gate is a rotation around the Z axis by $\pi/2$ and the **T** gate is a rotation around the Z axis by $\pi/4$. Performing two **T** gates back to back is equivalent to an **S** gate, and two **S** gates back to back is equivalent to a **Z** gate. The **H** gate is commonly used at the beginning of quantum algorithms to put qubits into a perfect superposition state.

Two-qubit gates commonly involve a control qubit and a target qubit. In this case, a gate is performed on the target qubit if the control qubit is in the $|1\rangle$ state. For example, the controlled-NOT, **CNOT** gate is a controlled-**X** gate. Two-qubit gates are used to create entanglement between the qubits.

Quantum gates must be *unitary*. This means they must be linear, reversible, and preserve the magnitude of the column vector. For example, the **X** gate is its own inverse, and if two **X** gates are applied sequentially, then the qubit returns to the original state. In other words, unitary operations coherently transform the quantum state. Conversely, measurements are non-unitary and irreversible. If a qubit in a superposition of $|0\rangle$ and $|1\rangle$ is measured and found to be $|0\rangle$, then it is then entirely in the state $|0\rangle$. This is an incoherent process, as the quantum state has effectively been destroyed, containing only classical information. Whether operations are unitary or not is important not only for quantum gates and measurements, but also for the noise that affects the quantum state.

Quantum states that are not pure are called *mixed states*. These states are combinations of pure states, each with an associated classical probability. Mixed states occur as the result of imperfect isolation and manipulation of the quantum state, which applies to all physically realizable quantum states. *Density matrices* are the equivalent of kets for mixed states. Density matrices can represent all pure and mixed states. A ket representation can be converted to a density matrix by taking the outer product of the ket with its conjugate transpose (adjoint)

$$\begin{bmatrix}\alpha\\ \beta\end{bmatrix} \longrightarrow \begin{bmatrix}\alpha\alpha^* & \alpha\beta^*\\ \beta\alpha^* & \beta\beta^*\end{bmatrix}, \tag{6}$$

where * denotes the complex-conjugate. A common metric to quantify the "quality" of a quantum state is the fidelity [48], which is defined as

$$Tr[\rho\sigma], \tag{7}$$

where the density matrix ρ represents the "correct" quantum state; σ, the actual quantum state; and Tr, the trace (diagonal sum) of the matrices multiplied. Simulating with density matrices allows one to keep track of classical probabilities and possible errors, in addition to the quantum transformations. This can be convenient in many cases [70]; however, the computational resources required increase significantly [13].

3 QUANTUM NOISE

Noise is present in all computing systems. However, it is quite a force to be reckoned with for quantum systems. In fact, noise is so pervasive that it is impossible to have a meaningful discussion about practical quantum computing without an in-depth consideration of its effects. Clearly, no benchmarking approach can succeed without considering noise and the resulting impact on measured or simulated results. This has been unfortunate, as quantum noise is difficult to

characterize and, in many cases, its effects are not well understood. Here, we provide a brief overview of quantum noise and the models used to represent it.

3.1 Physical Sources of Noise

Quantum noise can come from a variety of sources. There is never only a single source of noise in any given quantum system. Determining what the sources are and what their relative contributions are is a hard problem. Possible sources are unwanted interaction with the environment (both distinct events and inevitable decay of quantum states), unwanted interaction between qubits, or imperfect control operations. Each of these introduce error with significantly different characteristics, giving rise to different models. Here, we go over different common sources and discuss their physical impact. A summary of physical noise sources is provided in Table 1.

3.1.1 Interaction with the Environment. Qubits need to be perfectly isolated from the environment to maintain their state. If such a system could be constructed, then there would be no quantum noise. But no real system can be perfect, hence there is inevitably some interaction. This can be seen as a "measurement" of the system [12], as information is leaving the quantum state. As measurements are non-unitary, this kind of noise is also non-unitary. The expected amount of time a system can remain unperturbed is called the coherence time. Commonly reported are the T_1 and T_2 times. T_1 measures the expected loss of energy from the system; if a qubit is put into an excited $|1\rangle$ state, then T_1 is a measure of how long it takes to collapse to the $|0\rangle$ state. This is also called the qubit relaxation time [149]. T_2 measures the dephasing time; if a qubit is placed in the superposition state $|0\rangle + |1\rangle$, then T_2 determines how long it takes to polarize either to $|0\rangle$ or $|1\rangle$ [109]. Risk of interaction with the environment is increased when performing operations on the qubits, as the driving force of the operation comes from an external input. This is an unfortunate situation as two critical requirements have conflicting needs. The quantum state needs near perfect isolation to remain intact, yet also must interact with control mechanisms to perform useful computation. This is referred to as the coherence-controllability tradeoff [166].

Interaction with the environment can also produce unitary errors, such as global external fields that act on the qubits [11, 63, 85, 156, 157]. Such interactions can cause unitary rotations of the quantum state.

3.1.2 Interaction with Other Qubits. As previously mentioned, qubits can become entangled with each other. This means their states become correlated. While this is a frequently used tool in quantum computation, it needs to occur only when desired. If qubits interact accidentally, then this can lead to a mixture of their quantum states or decoherence [27, 117, 128]. This is referred to as cross-talk. This type of error has been particular difficult to characterize.

If left in perfect isolation, then cross-talk between qubits would lead to a unitary evolution of the state. Hence, all the information is still contained in the quantum state. However, the quantum state would be different than the one desired, which destroys the ability to manipulate it in a meaningful way. For example, we may wish to have two qubits that are far apart and highly entangled, to perform quantum teleportation.[3] However, if they interact with other, nearby qubits, then this will disperse and decay the entanglement [117]. If not in perfect isolation, then cross-talk can cause increased degradation of the quantum state. Say one is performing error correction, which involves interacting extra (ancilla) qubits with the qubits that store the data, and then measuring the ancilla to extract error information (syndromes). Cross-talk may cause the unintentional interaction of a

[3]Quantum teleportation is the transfer of quantum information between qubits by means of quantum entanglement in combination with a classical channel. It can be used to transfer information between non-adjacent qubits in a quantum computer or over long distances via a quantum network.

Table 1. Categorization of Physical Noise into Its Sources and Whether It Is Unitary or Not

	Environment	Other Qubits
Unitary	External Fields Over/under Rotations from Imperfect Control [12]	Cross-talk [27, 117]
Non-Unitary	Unintentional Measurements [12]	

data qubit with an ancilla qubit. Thereby some of the quantum information in the data qubit can get transferred into the ancilla. As a result, when the ancilla is measured, the computer would unknowingly extract information from the data qubit, corrupting its quantum state.

3.1.3 Imperfect Operations. Imperfect application of quantum gates can generate incorrect quantum states. Often, these are slight over- or under-rotations that are the result of imperfect calibration [26]. These kinds of errors do not directly destroy the quantum state but lead to coherent evolution of the quantum state into an undesired state [12]. This type of noise is predominantly seen in modern experiments [159] and its potentially catastrophic impact on the ability to perform error correction has been a concern in recent years [26].

3.1.4 Leakage. Many quantum systems that are used as qubits actually have more than two possible states. In this case, two of the possible states are selected to represent $|0\rangle$ and $|1\rangle$. In other words, the qubit is encoded in a subspace of a larger quantum system [164]. This subspace is called the *computational subspace*. It is assumed that the quantum systems remain in these two states (though other states may be used temporarily, such as in the implementation of two-qubit gates [30]). If a qubit unintentionally enters one of these other states, then it is referred to as leakage, and it can be particularly detrimental [49]. The return of a qubit back into the compuational subspace is called *seepage* [164]. Leakage and seepage can be either unitary or non-unitary, and can be caused by imperfect control or unwanted interactions with the environment [164].

3.2 Noise Models

Working at higher levels of the system stack, it is more critical to know how quantum noise will affect quantum operations. In this section, we transfer focus from physical sources of noise to the process of modeling them. The goal is to learn how noise disrupts the correctness of quantum algorithms during operation. As most research labs do not have their own physical quantum computer, and publicly available machines have low qubit counts, accurate and efficient noise models are greatly desired to facilitate simulation. Quantum noise is notoriously difficult to model accurately [159]. There are variations of quantum noise and it is possible that it may even be non-Markovian. Knowing what specific type of noise is present and how it affects a particular system is difficult to determine without extensive, physical experiments. However, there are a number of possible methods to estimate noise, with varying degrees of accuracy and computational efficiency. Realistic noise models are often intractable to simulate at scale, so simplifying assumptions are made to reduce complexity [99]. It is important to know when these assumptions are appropriate to make to produce realistic results. Here, we give a brief, high-level overview of different noise models and discuss their implications. A summary of noise models is provided in Table 2 and a summary of the physical noise processes they emulate is shown in Table 3.

3.2.1 Stochastic Pauli Noise. Stochastic Pauli noise is the simplest and most intuitive noise model. Additionally, according to the well-known Gottesman-Knill theorem [4, 59], it is easy to simulate using classical computers [26] and, at the same time, easy to correct using standard

Table 2. Categorization of Noise Models into Whether They Are Unitary and Whether They Are Efficiently Simulable or Not

	Efficient	Not Efficient
Unitary	NA	Coherent Rotations [26]
Non-Unitary	Stochastic Pauli Pauli Twirling [53] Clifford Channels [69, 71]	Amplitude/Phase Damping[149]

Table 3. Physical Noise and Corresponding Noise Models

Physical Noise Source	Noise Models
Interaction with Environment	Stochastic Pauli Noise Amplitude/Phase Damping Pauli Measurements
Imperfect Control	Coherent Over/Under Rotations

error correction procedures [159]. Hence, it has become popular [5, 79, 88]. It is most applicable for modeling unwanted interactions with the environment, which is effectively unintentional "measurements" of the quantum state [12]. It can be implemented by inserting an **X**, **Y**, or **Z** gate into a circuit at random with some specified probability. The effect on the overall fidelity can be estimated with Monte Carlo simulation [88]. Alternatively, representing the quantum state as a density matrix, ρ, the noise can be modeled as

$$N_i(\rho) = (1 - \epsilon_i)\rho + \epsilon_i^x \mathbf{X}\rho\mathbf{X} + \epsilon_i^y \mathbf{Y}\rho\mathbf{Y} + \epsilon_i^z \mathbf{Z}\rho\mathbf{Z}, \tag{8}$$

where ϵ_i is the total error rate on qubit i and ϵ_i^x, ϵ_i^y, and ϵ_i^z are the rates for each type of error, corresponding to the probabilities of inserting each gate [26]. This is also referred to as *depolarizing noise* [149]. If $\epsilon_i^x = \epsilon_i^y = \epsilon_i^z$, then it is called *symmetric depolarizing noise*. **X**, **Y**, and **Z** are operators performing the respective gate on the qubit. While **X**, **Y**, and **Z** gates are unitary operations (causing a coherent transformation of the quantum state), inserting them in a probabilistic manner does not represent a coherent process. Additionally, the linear combination of unitary operations, as in Equation (8), can represent a non-unitary operation. Hence, stochastic Pauli noise is an incoherent source [26].

A common strategy is to inject error only after each gate. However, this is not realistic as qubits can acquire error even when remaining idle [155]. Therefore, Pauli noise should be injected in every cycle. Numerous studies have found that stochastic Pauli noise models often lead to inaccurate and overly optimistic results [12, 26, 63, 69, 71, 106], but that they still can provide reasonable approximations in certain conditions. These include errors at the logical level under QEC [13, 26, 63, 69].

There are some natural extensions to this model that can result in more accurate simulations. Significant improvements can be made, while remaining efficiently simulable, by augmenting Pauli gates with Clifford group operators (i.e., Hadamard (**H**), phase (**S**), and **CNOT** gates) and Pauli measurements [71]. This involves the same process of inserting gates at random, but using a larger gate set, which, in addition to the Pauli gates (**I**,**X**,**Y**,**Z**), has **H**, **S**, and **CNOT** gates [100].

A fundamental problem with stochastic Pauli noise is that it is "not quantum enough" [26]. While the inserted Pauli gates are quantum operations, the choice of whether to insert them is based on a classical probability. While a classical noise model is familiar and intuitive from a computer

architecture perspective, it is not necessarily true depiction of the real errors occurring at the physical level.

3.2.2 Coherent Noise. Coherent noise models attempt to capture evolution of the quantum state that, while not destructive, is still undesired. One could see this as coherently performing a quantum program, just one that is different than intended. Physical coherent noise can be caused by imprecise classical control of the quantum operations [26], external fields, and cross-talk [63]. Modeling coherent noise can be difficult as some of the relevant sources of noise are not well understood. Hence, simplifying assumptions are made. While not exact, the goal is for the model to affect the quantum state similar to realistic sources. Examples include static Z-rotations [26], X-rotations in combination with Pauli-X errors [63], or rotations about a non-Pauli axis [71]. Coherent noise is not efficiently simulable, meaning the classical resources required to simulate grow exponentially with the system size. Hence, these simulations are limited to relatively small systems [12, 34, 90, 149].

Coherent noise is typically much more detrimental, with a much higher worst case error rate [12, 26, 159]. Additionally, many quantum algorithms consist of periodic circuits, where the same sequences of gates are repeated many times. Coherent noise is particularly harmful in this case, where its effects get amplified with each iteration [17, 18]. From this, it may appear that coherent noise is more important, and should be assumed unless known otherwise. However, this may not be true for all circumstances. The effects of coherent noise were analyzed on quantum error correction [26, 63]. It was found that the coherence of the error is reduced at the logical level, and is further decreased with a higher code distance. This means that it may be sufficient to assume stochastic Pauli noise at the logical level, even if the physical noise is coherent. However, it was noted that using a stochastic Pauli model for physical noise would significantly under estimate the error rate at the logical level.

Unfortunately, as modern quantum computers are not capable of QEC, they cannot make use of this resilience to coherent error. **Randomized Compiling (RC)** [159] is a novel approach that may help in this domain. The basic idea is to perform randomizing Pauli gates during the run of a quantum circuit, which are interleaved with the gates of the program. At each location that randomization is introduced, the previous randomization is undone to return the quantum state to the desired state. These randomizing operations disrupt the coherent noise and tailor it effectively into stochastic Pauli noise. Prior to execution, these additional randomizing gates can be fused with (compiled "into") the actual gates in the circuit. Depending on the gate set available in the system, RC can be performed with no overhead.

3.2.3 Amplitude/Phase Damping. **Amplitude Damping (AD)** is a non-coherent error model that captures energy loss from the quantum state into the environment, such as spontaneous emission of a photon [149]. This noise model is relevant to any quantum system with multiple energy levels, where there is an excited state and ground state, with a tendency for the excited state to decay to the ground state, such as ion-trap quantum computers that use the excitation levels of electrons. Additionally, the loss of energy must be to some environment, i.e., in the previous example, if the energy loss is a spontaneous emission of a photon, then there must be an environment for the photon to escape into. Hence, if the quantum system was *perfectly* isolated, then AD would not occur. AD is a realistic noise model but is also not efficiently simulable. However, models have been designed to approximate the effect of AD, but that remain simulable [71], such as Pauli Twirling [54, 129, 137, 149]. **Phase damping (PD)**, sometimes call pure dephasing, is equivalent to a phase flip channel [149]. This does not change the probability of a qubit being in either the $|0\rangle$ or $|1\rangle$ state, but it changes the phase between the two states. AD is closely related to the T_1 time and PD is closely related to the T_2 time, which are discussed in Section 3.1.1.

The nature of quantum noise greatly affects the quantum state, and by direct result, the performance of any potential quantum computer. Hence, even at higher levels in the system stack, one must give serious attention to the expected noise present in the system and be sure it is adequately accounted for. This impact of quantum noise is often overlooked or given secondary consideration. In many cases stochastic Pauli noise may be an overly simplified model. If so, then it will produce incorrectly optimistic results, especially for modern systems.

4 METRICS

Metrics enable quantitative analysis on the quality of a given system. Choosing metrics can be tricky, and can be misleading if not done wisely. This is true for classical systems and, unsurprisingly, is even more complicated for quantum systems. A number of quantum metrics have been developed and used widely; however, there is no unifying "gold standard" [55]. Depending on the use case, different metrics may be preferable.

A commonly used metric for quantum operations is the *process fidelity*. The process fidelity of a noisy quantum operation, $\tilde{G}$, relative to the noiseless operation, G, can be determined by [44, 48]

$$F(G(\rho), \tilde{G}(\rho)) = Tr\left[G(\rho)\tilde{G}(\rho)\right], \tag{9}$$

where ρ is the density matrix representing the input quantum state and Tr is the trace. At a high level, process fidelity measures "how similar" the noisy output quantum state is to the intended target. If the noisy operation $\tilde{G}$ is free of error, then the process fidelity will be 1. One drawback is that it is not strongly tied to a physical interpretation, when comparing two different mixed states [55]. However, if one of the states is a pure state (represented here by $G(\rho)$), and the other is a noisy mixed state (represented here by $\tilde{G}(\rho)$), then it is the overlap of the two states [55].

Another commonly used metric is the *average gate fidelity*, or conversely the *infidelity*, the average gate infidelity to the identity [13]. The fidelity of a quantum gate is its process fidelity, where the qubit's quantum state after applying the gate is compared to what it should be. This is complicated by the fact that the qubit could be in any possible state prior to the operation (anywhere on the surface of the Bloch Sphere in Figure 1) and the fidelity could be different for each state. A general solution to this is to average over all possible input states, i.e., to integrate over the surface of the Bloch Sphere [24, 31]:

$$\langle F \rangle = \frac{1}{4\pi} \int Tr\left[G(\rho)\tilde{G}(\rho)\right] d\Omega. \tag{10}$$

However, it was shown that it is also possible to achieve the same result by averaging over a finite set of inputs [24].

The *trace distance* [126] is an alternative that measures the distinguishability of two quantum states. It is defined as

$$D(G(\rho), \tilde{G}(\rho)) = \frac{1}{2} Tr\left[\sqrt{(G(\rho) - \tilde{G}(\rho))^{\dagger}\ (G(\rho) - \tilde{G}(\rho))}\right], \tag{11}$$

where again ρ is the density matrix for the input quantum state. $G(\rho)$ is density matrix of the ideal output quantum state, which is ρ transformed by the process G. $\tilde{G}(\rho)$ is the density matrix of the noisy output quantum state, which is ρ transformed by the noisy process $\tilde{G}$. $\dagger$ means the adjoint. The trace distance is related to the process fidelity via

$$1 - \sqrt{F(G(\rho), \tilde{G}(\rho))} \leq D(G(\rho), \tilde{G}(\rho)) \leq \sqrt{1 - F(G(\rho), \tilde{G}(\rho))}, \tag{12}$$

allowing bounds to be placed on one if the other is known. The diamond distance [51] is the maximum trace distance between any state and the state impacted by noise [19]

$$||G - \tilde{G}||_d = max_\rho \left(||(G \otimes \mathbf{I})[\rho] - (\tilde{G} \otimes \mathbf{I})[\rho]||_1 \right), \tag{13}$$

where [159]

$$||M||_p = \left(Tr(M^\dagger M)^{p/2} \right)^{1/p} \tag{14}$$

is the Schatten p-norm of M. Note that when $p = 1$, Equation (13) takes a form very similar to Equation (11). The diamond distance is a measure of the distinguishability of G and $\tilde{G}$ given a single use of either operator. This will be quite small for any well-performing hardware [19]. Here, ρ can be entangled with additional ancilla qubits, and these inputs are often the most sensitive [19]. The diamond distance is often used in proving fault-tolerance of quantum computers when doing rigorous analyses of errors in quantum circuits [6, 83, 135] as it represents the worst-case error. A drawback of this is that it may be overly pessimistic [154]. While there are no efficient known ways of computing it directly, there are methods of efficiently computing the bounds [154].

If one is interested in only the output measurement probabilities, then the *Hellinger fidelity* may be a more appropriate metric. The Hellinger fidelity is defined as $1 - Hellinger\ \ distance$. The Hellinger distance measures difference between two probability distributions [95]. For example, if we have two probability distributions p and q, and p_i and q_i are the probability of sampling i in each distribution, then the Hellinger distance is defined as

$$\frac{1}{\sqrt{2}} \sqrt{\sum_i \left(\sqrt{p}_i - \sqrt{q}_i \right)^2}. \tag{15}$$

If the distributions are identical, then the Hellinger distance is 0, and if there is no overlap, then the Hellinger distance is 1. The quality of a noisy quantum operation is quantified by determining the Hellinger fidelity of its measurement distribution to that of an ideal one. This is convenient in experiment, as all that is required is repeated iterations of the noisy operation and measurements in the standard Z-basis. It is provided as a standard metric on IBM's Qiskit [5].

Many other variants of distance metrics exist [95, 140]. We have provided introductory definitions to some widely used metrics, but refer the interested reader to comprehensive works for more in-depth discussion on which quantum metrics are useful and in which contexts [55].

5 QUBIT BENCHMARKING

Before talking about benchmarking a quantum computer, we have to talk about benchmarking the qubits themselves, even in a single (or two) qubit system. If one is constructing a quantum computer, then it is clearly of great interest how reliable the quantum operations (gates) are. The average gate fidelity coarsely predicts how many gates can be applied before the quantum state gets too corrupted. Additionally, quantum error correcting codes only work if the error is below a certain threshold [83]. On top of this, the overhead of the error correction strongly depends on the fidelity [111]. Quantum gates suffer from high error rates. Errors in classical switches could be less than 1 in 10^{15}, whereas effective quantum error rates are frequently above 1%, where the previously mentioned error sources and models apply and determining their impact is a complex task.

Unfortunately, experimentally determining quantum states, and the fidelity of quantum operations, is not straight forward. Measurements of a quantum system are destructive, so the state will need to be prepared or the operation performed for each measurement. In addition to this time overhead, errors in the initial **state preparation and the measurements (SPAM)** errors can obscure the error in the operation.

A brute-force approach to learn a quantum state is quantum state tomography [153]. This enables the classical extraction of all quantum information, and hence answers any questions we may have about its structure. This requires measuring a complete set of observables (physical properties that can be measured) that determines the quantum state. For example, say there is a single qubit in the quantum state

$$|\psi\rangle = \sqrt{0.8}\,|0\rangle + \sqrt{0.2}\,|1\rangle\,. \tag{16}$$

For numerical clarity we use a pure state; however, this process works for mixed states, as well. While this is a pure state, it can be described equivalently in density matrix form:

$$\rho = \begin{bmatrix} 0.8 & 0.4 \\ 0.4 & 0.2 \end{bmatrix}. \tag{17}$$

This state is unknown to the outside world, but the state can reliably be prepared by a quantum operation. The goal then is to determine this density matrix only by performing repeated measurements on the prepared state $|\psi\rangle$. All single qubit density matrices can be represented as a linear sum of the Pauli matrices [112]:

$$\rho = \frac{1}{2}\left(S_0\mathbf{I} + S_1\mathbf{X} + S_2\mathbf{Y} + S_3\mathbf{Z}\right). \tag{18}$$

Therefore, if we find the coefficients, then we can reconstruct the density matrix. These coefficients can be found by determining measurement probabilities when measuring along the X, Y, and Z bases [112]

$$\begin{aligned} S_0 &= P_{|0\rangle} + P_{|1\rangle} && = 0.8 + 0.2 = 1.0 \\ S_1 &= P_{|0\rangle+|1\rangle} - P_{|0\rangle-|1\rangle} && = 0.9 - 0.1 = 0.8 \\ S_2 &= P_{|0\rangle+i|1\rangle} - P_{|0\rangle-i|1\rangle} && = 0.5 - 0.5 = 0.0 \\ S_3 &= P_{|0\rangle} - P_{|1\rangle} && = 0.8 - 0.2 = 0.6, \end{aligned} \tag{19}$$

where $P_{|\phi\rangle}$ is the probability of measuring $|\phi\rangle$. Note that $S_0 = 1$ by construction, as $|0\rangle$ and $|1\rangle$ are the only two measurement outcome possibilities. S_3 only requires measurements along the standard Z-basis. To find S_1 and S_2, measurements need to be performed along the X and Y axes. This can be done with a basis change prior to measurement, which is done by applying the appropriate quantum gate just prior to measurement. For example, performing a Hadamard gate then a Z-basis measurement is effectively an X-basis measurement. Note that each measurement must be performed many times to achieve accurate probability estimates. The exact probabilities are shown in Equation (19). For the example we computed them directly with $Tr\left[\xi\rho\right]$, where ξ is the density matrix of the corresponding basis state. Once the coefficients are known the state can be described

$$\rho = \frac{1}{2}\left(1 * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + 0.8 * \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + 0 * \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} + 0.6 * \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\right) = \begin{bmatrix} 0.8 & 0.4 \\ 0.4 & 0.2 \end{bmatrix}. \tag{20}$$

This process can be generalized to multi-qubit systems as well, we refer the reader to a thorough introduction in Reference [112]. However, it does not scale well as it requires a number of measurements that is exponential in the system size. In addition to this, it is difficult to distinguish states with low probability from those with zero [84] and the computation to convert measured results into an estimate of the quantum state is intractable [38]. Hence, this approach is not feasible for large systems. There are notable variations of this process, such as Shadow Tomography [1]. While quantum state tomography has exponential cost, because it tries to answer all possible questions, shadow tomography attempts to only learn certain features by learning from measurements [76]. The name comes from the idea that one is not trying to learn the full density matrix, only the "shadow" it casts on the chosen measurements [1].

Quantum process tomography (QPT) uses the same approach as quantum state tomography, except that the goal is to identify a quantum operation, rather than the quantum state. Known input quantum states are generated and then the operation is performed. Quantum state tomography is then applied to the output states allowing identification of the process [37, 116]. As it follows the same procedure as quantum state tomography, it also requires exponential resources in the number of qubits. The key assumption that QPT makes is that the input state is known, and therefore only the quantum operation needs to be found. This simplifies the problem, and finding the parameters of the operation is equivalent to maximum likelihood estimation of a convex objective function (which has a single, global minimum) [62]. Hence, standard convex optimization techniques can be used to solve for it [25, 36]. This simplification comes at a cost, however. The input state is not necessarily known, as there can be errors in the state preparation process, which can lead to inaccuracy [62]. Compressed quantum process tomography [133] uses compressed sensing, a known classical signal processing strategy, to reduce the number of measurements required. The unitary operations performed by quantum computers can typically be represented by nearly-sparse process matrices, and hence can be well approximated by sparse matrices. These sparse matrices can be found with exponentially fewer measurements [133].

Direct Fidelity Estimation [48] is another clever method used to extract meaningful information without exponential overhead. This procedure estimates the fidelity of an arbitrary quantum state relative to a pure state. Here, the pure state is an error free state that is the intended result of a quantum operation; and the arbitrary state, what is actually produced. Note that complexity can be reduced by not attempting to learn everything about the state, seeking only to get the estimate of the fidelity of the arbitrary state. The estimate is achieved by measuring a constant number of Pauli observables (applying standard Pauli operations and then measuring). This is possible by making educated guesses about which Pauli observables are likely to reveal errors.

Gate set tomography (GST) is an extension of QPT [62, 103, 107]. Like QPT, the resources it requires increase exponentially with the number of qubits [62]. Hence, it cannot target large quantum systems and is also used mainly for 1- and 2-qubit systems. However, a significant advantage it has over QPT is that it is calibration-free, i.e., it does not depend on accurate descriptions of the initial prepared states [107]. This is significant, as QPT can generate highly inaccurate results when the gates used to prepare the input states have systematic error [103]. Like QPT, the process of finding the parameters of the quantum operations (gate set) involve maximum likelihood estimation of an objective function based on measurement results. However, the inclusion of SPAM errors in the gate set produces a non-convex objective function [62]. Hence, standard convex optimization techniques do not work, and a combination of approximate and iterative methods must be used. Another consequence of including SPAM errors is that it is not possible to characterize a single gate at a time, as in QPT. Rather, there is a minimal set of gates that must be estimated simultaneously [62]. The mathematical background of GST is provided in Reference [107] and the protocol is provided in Reference [62].

A scalable approach that has been utilized frequently in recent years is *Randomized Benchmarking* [40, 43, 84]. Randomized Benchmarking attempts to go beyond tomography by determining error probability per gate in computational contexts. Like GST, it is calibration free [107]. A further strength of randomized benchmarking is that is insensitive to variations in error between the different types of gates used [74, 104, 119, 158].

There are variations on the specific implementation [22], including numerous extensions [7, 52, 81, 99, 156]. We provide a high level description of the general approach, but note that different formulations may vary on the specifics. First, a random sequence of operations (gates) is generated. Commonly, gates are chosen uniformly at random from the Clifford set, as these can be efficiently performed on a quantum processor [96] and also can be efficiently simulated on a

classical computer [58]. The sequence is then appended with a final operation that undoes the action of the entire sequence. For example, if the sequence has a length of m and the operation (set of gates) applied at cycle i is denoted as c_i, then the effect of the first $m-1$ operations is

$$C = \prod_{i=0}^{m-1} c_i, \tag{21}$$

where $\prod$ denotes matrix multiplication. The final operation is then chosen to be the inverse, or adjoint, of C

$$c_m = C^\dagger \tag{22}$$

and hence the full sequence will become the Identity operation,

$$\prod_{i=0}^{m} c_i = c_m C = C^\dagger C = \mathbf{I}. \tag{23}$$

This final corrective operation is easy to determine, due to the Clifford set being efficient to simulate classically [96]. Hence, a quantum state subjected to this sequence will be returned back to its initial state (in the case of no error). This is often chosen to be the $|00\ldots0\rangle$ state. In experiment, the fidelity can be found by finding the probability of measuring the initial state [120]. Many sequences of the same length should be generated, and the results averaged over all cases. This process is then repeated, using sequences of various lengths. The fidelity will be a function of the sequence length and will drop exponentially as the the length increases. The fidelity can then be fit to a model, where constants will absorb the state preparation and measurement errors [96]. This enables the extraction of the average error per gate. For example, in a single qubit case with independent gate errors, the average fidelity of the sequence can be modeled as [97]

$$\langle F \rangle = A(-2r+1)^m + B, \tag{24}$$

where r is the average gate error and A and B are the fitting constants. This enables r to depend exclusively on the fidelity of the gate operations [74]. This process can be applied to many qubit quantum states with relative ease, as the resources required scale polynomially in the number of qubits [62, 97, 98]. As a result, it has become a standard approach to benchmarking quantum systems. The common practice of using the Clifford set, while convenient, is also a limitation, as by itself it is not universal [62]. However, strategies have been found to extend randomized benchmarking beyond the Clifford set [33, 74, 81]. Additionally, note that the scalability of randomized benchmarking is enabled by only seeking a subset of the quantum information, it does not provide the full tomographic information about the gates [45, 103].

Modern analog quantum computers have been able to simulate systems that are hard even for classical supercomputers [29, 80, 138, 163]. This raises the concern of validating their output, as they may be untrustworthy [57, 73] and it is challenging to classically certify them [163]. *Hamiltonian Learning* is a process by which the Hamiltonian of a system is estimated to validate that it is simulating the correct dynamics [10, 45, 61, 162, 163]. Additionally, reconstructing the Hamiltonian of a quantum system will provide detailed diagnostic information that can enable noise reduction in experiment [45]. The Hamiltonian of an n-qubit system, which has dimension $d = 2^n$ can be described by d^2 parameters, though most Hamiltonians of interest can be described by $m = O(poly(n))$ [45]. Utilizing known information about the system can significantly reduce the number of measurements required and make the process tractible [61]. *Compressed quantum Hamiltonian learning* [161] is a process in which the dynamics of subsystems of a large device are measured against the dynamics of a smaller system, enabling a model to be created for the larger

system. Using a *trusted simulator* [162, 163], which has a firm known mathematical model, enables an absolute model to be generated for the larger system [161].

6 (QUANTUM) COMPUTER BENCHMARKING

This kind of benchmarking is more similar to classical benchmarking. The idea is to determine the compute capability of a quantum system running a program. This is sometimes referred to as *holistic benchmarking* [107, 108]. Note that this is still considering near-term computers that do not use error correction. It shares a number of considerations with classical computing, such as latency and available parallelism. However, these metrics are not as informative for quantum computers. As to be expected, there are a number of additional considerations. When discussing benchmarking for full quantum computers, it is important to reconsider the role of quantum error and, if applicable, quantum error correction. As previously mentioned, quantum noise is not equivalent to classical noise, and these differences get more pronounced the larger the system is.

In classical computing it would be feasible to obtain an error rate per gate and stitch these error rates together to generate an error model for a larger circuit. The quantum equivalent would be finding a fidelity for each gate, and then assuming this error rate for each gate on each qubit in the system throughout the entire program. This is *not* accurate for quantum circuits as quantum noise is context dependent [155]. Even if good estimates of gate fidelity can be obtained, then using this information to model larger systems with more qubits is not straightforward. A gate performed on one qubit may induce error in another, via quantum entanglement or physical proximity. Hence, qubits must be considered as a monolithic system and their error rates cannot be considered independently [44, 47]. This is problematic as it makes it difficult to understand how noise affects large quantum computers. The whole point of creating large quantum computers is to create states that cannot be efficiently classically simulated. Unfortunately, that also means the noise becomes impossible to simulate. Hence, accurately characterizing the noise, and its significant effects on the reliability and performance, is not straightforward.

This has a few key impacts on benchmarking quantum computers. One is that it intensifies the error that is introduced when using a reduced program size as a benchmark. The error rates per qubit or per operation may be higher on a larger system. Some experimental evidence has shown that this may be overly pessimistic [44], but increased system sizes will no doubt increase susceptibility to noise due to complexity [101]. Hence, the rate of success of a program on a small system may be significantly different from that of a larger system, and extrapolating results is not straightforward.

6.1 Program Benchmarks

An intuitive approach is establishing a set of programs and measuring the performance of a computing system performing each one. As previously mentioned, this is common practice for classical computers. Replicating this for quantum computing would be collecting a set of quantum programs that are representative of the algorithms we would like to run on them. Common examples may be Shor's [136] for prime factorization or Grover's [66] for unstructured search. There are intuitive advantages to this approach, particularly in making the system perform a "real-world" task. IonQ, a quantum start up that has an 11-qubit ion-trap quantum computer, appears to favor this approach. They tested their computer on the Bernstein-Vazirani [14] and Hidden Shift [125, 152] algorithms, and their metric for performance was the likelihood of measuring the correct output [165]. They claim that these algorithms are representative benchmarks and the results proved their system was the best as of early 2019.

A challenge that this approach introduces is that someone must decide which programs are important. This introduces the issue of invested interests [91]. In the classical computing domain, a

lot of money is on the line when benchmarking hardware [75]. This problem is exacerbated by the fact that different, competing quantum technologies are superior at different programs. For example, the program benchmark approach was used in Reference [92], where a handful of small quantum circuits[4] were used to benchmark and compare the performance of an Ion-Trap quantum computer with a superconducting quantum computer. Amongst the chosen benchmarks were three-qubit circuits implementing the Toffoli and Margolis gates. It was found that the higher connectivity (ability of different qubits on the machine to interact) of the ion-trap computer allowed it to have a much higher relative success rate on the Toffoli circuit, which contained more two-qubit gates during the program. The success rate was more comparable on the Margolis circuit, which required fewer two-qubit gates. The authors note that how well the quantum architecture matches with the requirements of the algorithm is a major determinant of the performance [92]. So, which of the two benchmarks are more insightful? This question highlights the difficulty of creating a representative set quantum program benchmarks. While quantum computing is more complicated, much can be learned from previously discovered pitfalls and fallacies of classical program benchmarking. Intuitively, the benchmarks will be set by the customer, rather than the manufacturer. To the extent possible, the benchmarks should reflect the applications that customers will be running on them. Another key component of accurate (and honest) benchmarking with a set of programs is transparency [75]. This means providing data on all the programs and not unfairly weighting some results over others. For example, a quantum benchmark suite could be created that contains many circuits that have few two-qubit interactions, but also a few prominent circuits that require many two-qubit interactions, such as the **Quantum Fourier Transform (QFT)** [136]. A quantum computer with low connectivity could be benchmarked on all circuits, with only the average performance being reported. Such analysis could falsely inflate perception of the performance.

While impressive, these modern computers are very small compared to the computers we hope to build in the coming years. Hence, these benchmarks are also very small compared to truly useful programs. While running smaller versions of real-world applications introduces error, and is accepted in classical benchmarking, this is exacerbated for quantum computers. Entirely new issues may be introduced when scaling up and it is difficult to say whether measurements taken today are good indicators of future performance. For example, IonQs computer [165] has all 11 qubits fully connected, meaning each qubit can directly interact with every other qubit. This configuration is possible at this scale, but may not be for a system with hundreds or thousands of qubits. Such a system will likely require multiple fully connected groups of qubits and communication will need to be orchestrated between them [101]. This introduces additional complexity that is not found in these small-scale benchmarks. This is analogous to the classical benchmarking of machine learning inference accelerators. An accelerator that performs well on MNIST [87] digit recognition will not necessarily perform well on 1000-class ImageNet [41] classification. While the problem and computation is similar, ImageNet requires much more data, and memory management becomes the bottleneck.

A more fundamental question is what quantum programs will be useful in the future. Famous algorithms such as Shor's [136], Grover's [66], and quantum chemistry [32, 102, 110] are obvious examples. However, for the most part, these algorithms will remain well out of reach for some time. Currently, classical-quantum hybrid algorithms [46, 114, 130, 160] are popular due to their ability to make use of the limited resources of modern quantum computers. It is important to remember that quantum algorithm design is still an emerging field, and what actually is the best use

[4]In literature on quantum computing, circuit refers to a sequence of quantum gates (instructions). It is analogous to "program" in classical computing.

of quantum computers is still unknown. This presents a moving target, which suggests quantum research should not too heavily invest in any one direction [20].

6.2 Quantifying Capability

Considering the current limitations of quantum computers, it may be more insightful to focus on how much work a quantum computer is capable of, in contrast to performance results on specific algorithms. Because quantum technology is not mature, modern quantum computers are not yet capable of performing commercially useful algorithms. However, creating larger, more functional computers is of great interest, regardless of the applications they perform. This type of benchmarking is a shift away from traditional, classical benchmarking, and is an attempt to uniquely and objectively quantify the "capability" of modern quantum computers. The aim is to abstract out as much as possible, such as unique architecture characteristics and performance on specific algorithms, and create a simple metric that indicates the general computational power of the machine. While this benchmarking approach can be applied to all modern quantum computers, it is intrinsically tied to the concept of *quantum supremacy*—demonstrating that a quantum computer can perform work that is not possible with classical computers. Hence, this type of benchmarking is not just used as a comparison between different points in the design space of quantum computers, but is also the experimental assessment of the capabilities of quantum computers with respect to classical machines. Such benchmarks seek practical demonstrations of quantum computers solving well-defined problems, which are also intractable (in terms of time and hardware resources required) for classical computers. The best known classical algorithm/implementation should be used as a baseline for comparison in this case, while accounting for all sources of noise on the quantum side.

The importance of demonstrating quantum supremacy cannot be overstated. A central motivation for creating quantum computers is to solve problems that cannot be solved by other means [146], which will only be possible if quantum computers can achieve (and go beyond) quantum supremacy. It has been argued that this will be fundamentally impossible due to noise [78]. Hence, even if an experimental demonstration of quantum supremacy does not provide scientifically or commercially useful results, then it is an invaluable proof of concept for future research efforts. The success of quantum computing research is contingent upon reaching this goal. The key question then is how to quantify capability of quantum machines accurately and how to be sure when they truly achieve quantum supremacy.

Cross-Entropy benchmarking [21] can be used to validate the output of a quantum computer and was created specifically as method to test for quantum supremacy. The previously described program benchmarks consider decision problems, where the measurement at the end provides an answer to a specific question. In significant contrast, Cross-Entropy benchmarking considers *sampling problems*. Here, the measurement at the end effectively allows for the sampling from a certain distribution. It is the ability of the quantum computer to create such a distribution that acts as the demonstration of quantum supremacy. Hence, it must be shown that the quantum computer produces the distribution with a sufficient fidelity and that a classical computer (using polynomial resources) cannot provide a sampling from that same distribution. Beyond demonstrating supremacy, validating sampling problems is important, because they correspond to important quantum applications, such as quantum simulation.

To produce the output distribution, a quantum circuit is constructed by choosing quantum gates (from a universal set) at random. Using a random circuit allows for limited guarantees of computational hardness, as it does not have structure [9, 21, 23, 28]. Executing this circuit on a quantum computer will produce a quantum state in which some states are much more likely than others,

yet is also widely distributed over the 2^n possible measurement outcomes, where n is the number of qubits. The distribution resembles a speckled intensity pattern produced by light interference in laser scatter [9]. For a large quantum computer, 2^n will be much greater than the number of samples (executions of the circuit) that can be taken. Hence, it is very unlikely that two different measurements will be same [21]. This makes it difficult to discern it from a uniform random number generator by simply looking at the samples. However, the output can be distinguished and validated if the circuit is also classically simulated [57, 93, 115, 150]. The classical simulation (which requires exponential resources on a classical computer) will provide the exact output quantum state, and hence also the exact probabilities of measuring each result (in the absence of noise). This allows the comparison of the output of the experimental quantum circuit with the ideal quantum circuit, and also with any classical algorithm that attempts to generate the same distribution. The information theory definition of entropy is [134]

$$H(X) = -\sum_{i=0}^{N-1} P(x_i) log P(x_i), \tag{25}$$

where X is a random variable that can take on N values and $P(x_i)$ is the probability of observing $X = x_i$. The output of the quantum circuit, and a classical algorithm attempting to emulate it, is a random variable that can take $2^n = N$ values. The cross-entropy between the distribution of the ideal quantum probabilities (p_U) and the polynomial classical algorithm probabilities (p_{pcl}) can be defined as [21]

$$H(p_{pcl}, p_U) = -\sum_{j=1}^{N} p_{pcl}(x_j|U)\ log\ p_U(x_j), \tag{26}$$

where $p_U(x_j)$ is the probability that the quantum circuit, U, will produce the output x_j and $p_{pcl}(x_j|U)$ is the probability that the classical algorithm emulating U will also produce x_j. The value of interest is the average quality of the classical algorithm, which can be found by averaging the cross-entropy over an ensemble of random quantum circuits, $\{U\}$:

$$E_U[H(p_{pcl}, p_U)] = E_U\left[\sum_{j=1}^{N} p_{pcl}(x_j|U)\frac{1}{log\ p_U(x_j)}\right]. \tag{27}$$

It was argued in References [21, 23, 28] that the polynomial classical algorithm cannot accurately reproduce the distribution.

When performing the circuit on a quantum computer, a single error (such as X or Z error) will cause the output to become nearly statistically uncorrelated with p_U. Hence, depolarizing noise will cause an increase in the entropy of the output of the quantum computer, and it will resemble that of a uniform distribution. This means that the quantum computer, which is subject to noise, will only barely be able to produce a superior cross-entropy to that of a uniform random number generator,

$$E_U[H(Uniform, p_U)] = log\ N + \gamma = H_0, \tag{28}$$

where $\gamma = 0.577$ is the Euler constant. On this concept the quantum supremacy test is based. Any classical or quantum algorithm, A, which produces bitstring x_j with probability $p_A(x_j|U)$, can be evaluated on how well it predicts the output of an ideal quantum random circuit, U, by comparing its cross-entropy relative to that of a uniform classical sampler. This metric is called the cross-entropy difference [21]:

$$\Delta H(p_A) = H_0 - H(p_A, p_U) = \sum_j \left(\frac{1}{N} - p_A(x_j|U)\right)\ log\ \frac{1}{p_U(x_j)}, \tag{29}$$

If the algorithm A produces the distribution perfectly, with no errors, then the cross-entropy difference will be 1. If the algorithm A produces an uncorrelated distribution, then the cross-entropy difference will be 0 [9, 21]. Effectively, this is a measure of how consistent the outcomes are with the predicted probabilities [18]. A classical algorithm (using polynomial resources) should fail, and a quantum algorithm running on a sufficiently powerful quantum computer should succeed. The experimentally determined cross-entropy difference, denoted by α, can be found by taking m samples from the quantum computer, with each producing a bit string x^{exp}, and then using classical simulation to find the value $p_U(x^{exp})$. α is then estimated by [21]

$$\alpha = H_0 - \frac{1}{m}\sum_{j=1}^{m} log\frac{1}{p_U(x_j^{exp})}. \tag{30}$$

Quantum supremacy is achieved if a quantum computer can produce a higher α than a classical computer can. Unfortunately, quantum supremacy also implies that the p_U values required to determine α, which are produced by classical simulation (using exponential resources), can no longer be computed by a classical computer in a reasonable amount of time. Hence, measuring α directly is not possible if the quantum computer truly has achieved quantum supremacy. However, it should be possible to extrapolate α if it has been reliably found at sizes that are slightly below the limit of quantum supremacy (p_U can still be computed by sufficiently powerful classical supercomputers [21]). In the summer of 2019 Google used cross-entropy benchmarking[5] to quantify the capability of their 52-qubit quantum computer, Sycamore, where they claim to have demonstrated quantum supremacy [9]. It is also possible to use cross-entropy benchmarking to find the average fidelity of individual gates [9, 21], similar to randomized benchmarking.

Note that tests for quantum supremacy doesn't necessarily require a universal set of gates on the quantum computer. For example, in the case of Boson Sampling, recent work has demonstrated that these algorithms can be composed by notably more noise-tolerant (yet not necessarily universal) set of gates [151].

A similar and influential sampling-based benchmark is quantum volume from IBM [16, 39]. The process of determining quantum volume shares many of the steps with cross-entropy benchmarking. According to the authors [16], there are 4 factors that determine the capability of a quantum machine:

(1) The number of physical qubits
(2) The number of gates that can be applied before errors make the the output unreliable
(3) The connectivity of the machine
(4) The number of operations that can be run in parallel

It is assumed that quantum volume targets modern, noisy quantum computers. Hence, factor 2) is referring to running a quantum algorithm without error correction and it is assumed there is an upper limit on the number of gates possible. In the future, it will be of more interest to determine whether the error rate is low enough to enable efficient error correction, or how often error correction needs to be applied. Hence, quantum volume will need to be adapted or superseded in the future [39].

Similar to cross-entropy benchmarking, quantum volume attempts to abstract out all considerations and generates a single number that quantifies the capability of a quantum computer. The idea is to measure something that can be improved by each of the four considerations, meaning

[5]Google used linear cross-entropy benchmarking, which differs from the standard cross-entropy benchmarking process explained here. However, the argument is similar for why linear cross-entropy benchmarking is easy for a quantum computer and hard for a classical computer.

systems that are superior in each consideration will generally achieve a higher quantum volume. The score is determined by the largest random quantum circuit a quantum computer is able to complete successfully. Whereas cross-entropy benchmarking uses the cross-entropy to quantify the validity of the output, quantum volume uses the heavy output generation problem [3]. Heavy output generation also requires classical exponential time to validate the output from the quantum computer, as it relies on the ability to classically simulate the quantum operations [3]. Again, this will not be possible once physical quantum computers reach the scale where their quantum states are not efficiently classically simulable. By simulating the quantum operation, the output measurement probabilities can be exactly determined. When ordering all possible outputs from most likely to least likely, all outputs that have a greater probability than the median are called *heavy outputs*. The quantum circuit is then repeatedly performed and the output measured. If the measurements produce *heavy outputs* more than 2/3 of the time, then the quantum computer "passes"; otherwise, it fails. If noise has completely destroyed the information in the quantum state, then heavy outputs will be generated 50% of the time [39]. The largest quantum circuit that the computer can get a pass on corresponds to the quantum volume, V_Q [39],

$$log_2 V_Q = argmax_m \, (min(m, d(m))), \tag{31}$$

where m is the number of qubits and $d(m)$ is the maximum achievable depth of an m-qubit circuit. Hence, quantum volume is the largest square ($m = d$) circuit that can be successfully run.

Quantum volume has some limitations. A number of which were addressed in Reference [18], which proposes a framework called Volumetric Benchmarks. The idea is to make quantum volume more general, by allowing different shapes of circuits, kinds of circuits (random, periodic, subroutines of algorithms), and different criteria for success. As noted by the authors of Reference [18], errors will affect different kinds of circuits differently, such as coherent errors getting magnified by periodic circuits but getting smeared out by random circuits. This provides evidence that universal benchmarks should be avoided. While quantum volume is a novel and useful concept, it uses exclusively random circuits. As we show in our simulations, random circuits have a similar affect to applying randomized compiling [159]. Another potential drawback is that quantum volume can produce potentially misleading results. For example, a Honeywell quantum computer achieves a high quantum volume score due to its extremely high fidelity operations [143]. However, given the relatively low qubit count, its dynamics can be simulated classically with ease [2].

Another significant approach is that of Cycle Benchmarking [44]. This is similar to the process of randomized benchmarking. The core idea is to break a quantum program into sets of operations on all the qubits (cycles) and then individually characterize the fidelity of each cycle. This allows one to quantify how well the computer can do specific operations. Additionally, arbitrary programs can be broken into a finite set of cycles [155]. This prevents the number of required characterizations from growing exponentially with the number of qubits. Using cycle benchmarking, the "benchmarks" would be individual cycles and the metric is the process fidelity, as defined in Equation (9).

7 FAULT-TOLERANT (QUANTUM) COMPUTER BENCHMARKING

Thus far we have considered qubit benchmarking and holistic computer benchmarking for near-term quantum machines. While there has been much effort to create practically useful quantum algorithms for these machines, as of yet their performance falls far short of their classical competitors. Hence, the main goal of these near-term demonstrations has been for research and to learn how to achieve scalability. Once computers are built that have sufficient qubit counts and sufficiently low error rates, full-scale QEC [62] will become possible. QEC can detect and correct

errors during the run of a program, and hence enable computers to perform arbitrarily long quantum programs. This is referred to as fault-tolerant quantum computing [118]. Such computers will be capable of solving real-world problems with much greater speed than classical computers, performing well beyond the threshold for quantum supremacy. Naturally, such fault-tolerant computers will significantly change the landscape of benchmarking as it breaks many of the assumptions of previously described techniques. For example, the maximum circuit depth is effectively infinite, hence a metric such as quantum volume would be limited only by qubit count. More fundamentally, the quantum states of these machines would be much too large to simulate, and hence the classical validation of the heavy output generation problem would not be possible. A typical use case for large-scale fault-tolerant quantum computers will be running algorithms that will be impossible to run on a classical computer, but for which the output can efficiently be checked for correctness on one [94], such as Shor's algorithm [136].

However, no fault-tolerant computers have been built to date, and hence there is not much research dedicated to benchmarking them. Likely, quantum benchmarking will look much more similar to classical benchmarking, where the emphasis will transition to performance (latency to produce the final result) amongst quantum processors that have sufficiently many qubits to perform the algorithm of interest. Benchmarking may take a similar form as program benchmarks, as described in Section 6.1, except the chosen benchmarks would be real-world, full-size applications rather than sample toy cases. Additionally, unlike the modern computers performing the program benchmarks described in Section 6.1, a fault-tolerant computer should not fail due to noise. Hence, the probability of success will not be a representative metric.

The performance of fault-tolerant computers will have a strong dependence on the method QEC used as it has a high overhead. QEC utilizes many qubits [50], involves many, repeated quantum operations, and also requires significant classical hardware resources to manage its orchestration [105]. Both the qubit chip [42] and the supporting classical architecture [147] will need to be specifically designed to support QEC. For an introduction to quantum error correction, we refer the interested reader to Reference [60].

8 SIMULATIONS

To illustrate the impact of the different types of quantum noise in different computational contexts, and the resulting difficulty of choosing representative benchmarks, we run representative key quantum algorithms at sizes that are experimentally feasible. All benchmarks used are listed in Table 4. We use a Quantum Adder [35], the QFT [136], and the **Quantum Approximation Optimization Algorithm (QAOA)** [46]. In addition, we use an idle circuit (which has no computational gates) and a random circuit (composed of random X, Y, Z, H, and CNOT gates), for reference. Note that this is not the same randomized circuit as used in Quantum Volume [39], which generates a set of arbitrary random unitary operations, which need to be decomposed into a universal gate set. For our random circuit, we want to view the effects of performing our gate set in random fashion, without introducing the complexity of gate compilation (which is needed for our other algorithms). Unless otherwise stated, our metric is the process fidelity [48] described in Equation (9). We note that other metrics may be equally suitable. However, we chose process fidelity as it is widely used in the quantum information science community [55] and it shines light on the complexities of benchmarking mentioned earlier.

The QFT and QAOA benchmarks contain gates that are precise rotations around the X- or Z axis of the bloch sphere. $R_z(\theta)$ is a rotation around the Z axis by angle θ and $R_x(\theta)$ is the equivalent for the X axis. Currently available modern quantum computers, such as IBM's machines [5], can perform these operations directly. This can be powerful, as it enables quantum computation to

Table 4. Benchmarks Used in Simulations

Benchmark	# Qubits	Logical Depth	Physical Depth	Gate Set	Metric
IDLE	4	2-70	2-70	I	Process Fidelity
RANDOM	4	2-70	2-70	I, X, Y, Z, H, CNOT	Process Fidelity
Adder	7	30	30	H, T, CNOT	Process Fidelity
QFT	4	10	10	H, R_z, CNOT	Process Fidelity
QFT (Clifford+T)	4	10	229	H, T, S, CNOT	Process Fidelity
QAOA	8	10	10	H, R_z, R_x, CNOT	Expectation Value
QAOA (Clifford+T)	8	10	113	H, T, S, CNOT	Expectation Value

Qubits is the number of qubits required for each input size. Logical depth is the depth of the quantum circuit (number of sequential gates) required before compilation into Clifford+T set. Physical depth is the depth after compilation. The Adder and QFT have two different input sizes.

occur relatively quickly, which also mitigates the effects of noise. As this gate set is likely to be used for some time into the future, we include it in our simulations of the QFT and QAOA. However, this gate set is not compatible with RC or quantum error correction. Hence, it is not scalable to the level of fault-tolerant quantum computers. For progressing into the fault-tolerant regime, the Clifford+T set is a good option. This gate set is universal and enables use of RC. Hence, we include Clifford+T versions of the QFT and QAOA as well. While we are using the Clifford+T gate set, we are not performing error correction in the simulation. Hence, these do not represent fault-tolerant computations. However, using this gate set can provide insight on how the impact of noise is affected by the chosen gate set. Each gate in Clifford+T can be implemented with the previously mentioned X- and Z- rotations – using the Clifford-T set just restricts the angles used. When performing these gates on physical qubits, as we do here, the operations are not changed in any fundamental way. While we do not show simulations of error correction, it should be noted that, to perform error correction, groups of physical qubits would form a single logical qubit, and the logical operations consist of many individual gates on the physical qubits [50].

The drawback of Clifford+T is that precise rotations must be broken into sequences of gates that can perform the operation approximately. This increases the length of the circuit. For single qubit gates, we use the gridsynth decomposition method from [132] that finds an approximation to Z rotations with Hadamard (H), T and S gates. This is sufficient, as any single qubit gate U can be implemented using the Euler angles [132]

$$U = R_z(\beta)\, R_x(\gamma)\, R_z(\delta) = R_z(\beta)\, H\, R_z(\gamma)\, H\, R_z(\delta) \tag{32}$$

for some β, γ, and δ. Hence, any quantum gate can be approximated by a sequence of approximate R_z gates and H gates. An additional complication exists in that controlled versions of these gates cannot be implemented directly [82]. For a general case, controlled 2-qubit versions of the gates can be implemented by including an additional ancilla (scratch) qubit and using an alternative sequence of gates [8]. However, both the QFT and QAOA can be implemented with just CNOT gates and single qubit R_z gates. Hence, controlled- H, S, and T gates are not required.

The Quantum Adder performs binary addition using a sequence of quantum full-adders on two input integers that are basis state encoded (1 qubit for each bit). If performing n-bit addition, then the quantum adder requires $3n + 1$ qubits. We perform 2-bit addition (using 7 qubits). The adder only uses gates in the Clifford+T set and hence does not require gate decomposition. It does use the Toffoli (doubly controlled X gate), which we implement with a sequence of CNOT, H, and T gates.

The QFT effectively performs the Discrete-time Fourier Transform on the amplitudes of the input quantum state, and is the core of Shor's algorithm. The width and depth of the circuit are determined by the number of input qubits. We use 4 input qubits. The circuit contains Hadamard gates and controlled-Z rotations, which can be implemented with CNOT and R_z gates.

QAOA has gained a lot of attention recently as it is considered to be a strong candidate to demonstrate quantum supremacy. QAOA is a *variational* algorithm, where quantum computation is used in tandem with classical optimization [160]. It is commonly applied to the Max-Cut problem. The input to Max-Cut is a graph that represents a combinatorial optimization problem. Vertices are variables and the edges between vertices are constraints. The goal is to partition the vertices into two sets, maximizing the number of edges between the sets. The *maximum cut* is the number of edges between the sets in a best possible partition, which is the most constraints that can be satisfied simultaneously. Max-Cut is NP-Hard, however, it is *not* expected that quantum computers will be able to solve NP-Hard problems in polynomial time. QAOA only provides an approximate solution and currently has worse performance than classical approximate algorithms [56, 86]. QAOA consists of a sequence of stages. Each stage consists of controlled-Z rotations (determined by the input graph) followed by single qubit X rotations. The hyperparameter p determines the number of stages in the quantum circuit. For each stage, there is a parameter γ, which determines the angle of the controlled-Z rotation, and a parameter β, which determines the angle of the X rotations. The main challenge of QAOA is to determine an optimal set of parameters to produce good output [67]. When implemented on a real quantum computer, optimization algorithms such as SPSA [139] or gradient descent can be used to update the parameters [144]. This requires many optimization passes, where each pass requires many (thousands or millions of) samples of the quantum circuit output. The number of samples required heavily depends on the problem size and noise level in the system. Here, we consider only a pre-optimized circuit with $p = 1$. When $p = 1$, the optimal circuit is guaranteed to produce an output with an expectation value that is at least 0.6924 times the maximum cut [46, 65] on 3-regular graphs. Here, we use a 3-regular input graph with 8 vertices (represented by 8 qubits), which has a maximum cut of 12. This means QAOA should produce an expectation value of 8.3 in the case of no error. For this problem, we use expectation value of the output as the metric. As QAOA is designed to also work on a small scale, it can be used effectively with continuously parametrized rotation gates available on near term machines. We implement QAOA with these gates as well as with the Clifford+T set. A useful tutorial for QAOA is provided in Reference [113].

For our simulations, we assume the quantum computer has an all-to-all connectivity and full parallelism. This means 2-qubit gates can be performed without any overhead for movement, and multiple single qubit gates can be performed simultaneously, given that they do not operate on the same qubits. Moving qubits with swap gates to compute on machines with limited connectivity is a well-studied problem [89, 167]. Our observations here will also apply, but there will be further complicating factors depending on the specific topology. For discussion on how noise and topology interact see Reference [148].

We also view the impact of RC [159] in each of our simulations. RC was designed as a method to convert coherent noise effectively into stochastic Pauli noise. This is significant, as coherent noise can be much more destructive and is predominantly seen in physical experiments. Hence, RC not only provides a significant noise mitigation technique, but also maintains the validity of previous theorems and results that have been generated by assuming stochastic Pauli noise. We emphasize, as noted by the original authors [159], that RC is not designed to have any impact on a Pauli noise model, hence we expect to see no improvement in the fidelity if a Pauli noise model is used. Additionally, for most of our simulations we are using process fidelity as the figure

of merit, for which RC is not expected to add as much benefit. RC will provide greater impact if using a norm-based measure, such as the trace distance in Equation (11). To perform RC, we need to divide the gate set into "easy" and "hard" gates. Easy and hard can generally be thought of as the difficulty in implementing the gate, such as the expected error rate, but the essential requirement is that the physical noise on the easy gates be independent of which easy gate is performed. We follow [159] and set easy gates as the Pauli gates and the phase gate S; where the hard gates are H, T, and all 2-qubit gates. However, instead of controlled-Z, we use CNOT. Nearly all gates in the decomposed circuits are "hard" gates. Hence, it is necessary to interleave these gates with idle cycles to implement RC. From a high level, simulation perspective, this initially seems to imply that the cost of RC is a near doubling of the circuit length. However, as noted by the authors of Reference [159], on real hardware (such as ion traps and superconductors) entangling operations, which are required for "hard" multiple qubit gates, must be inserted between local gates. These local gates, which are required even without RC, can be "compiled into" the randomized gates. Hence, in practice RC can be implemented with no additional circuit length overhead. Therefore, to make our simulations more representative of physical experiments, we insert the additional idle operations into our circuits whether RC is performed or not.

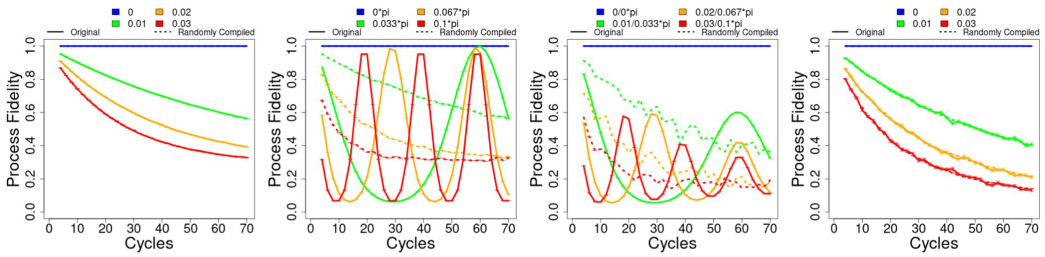
We use four representative noise models that fall into different categories discussed in Section 3. While no quantum system will have a single source of noise, we use them in isolation to demonstrate how the nature of the noise (along with the assumptions made in noise models) significantly impacts the results. The first is a standard Pauli noise, where the probability of X, Y, and Z errors are all equally likely. While the most commonly used noise model, it generally provides the worst (and overly optimistic) estimates or error correcting threshold error rates [69]. Pauli noise is a non-unitary and incoherent model. To model purely coherent and unitary noise, we follow the same approach as in Reference [26], where we assume constant Z-rotations by angle θ for each qubit, $(e^{i\theta Z})^{\otimes n}$, for various values of θ. While this model makes some simplifying assumptions, it is representative. The third is a combination of Pauli and coherent noise, where we follow the model in Reference [63]. This includes static X rotations and X Pauli errors. The fourth error model is Amplitude Damping, which is a commonly used and realistic noise model. It models loss of energy from the system to the environment and is a non-unitary process. We sweep the noise over a range of values that are similar to experimental error rates and are expected in near term computers, as listed in Table 5. Each noise type is injected in every qubit in every cycle, regardless if the qubit is operated on or not. We assume the same noise rates for single and two-qubit gates. While two-qubit gates will typically have a higher rate of noise in a physical experiment, our values are swept over ranges typically seen for both single- and two-qubit gates. Unless otherwise stated, our metric of choice is the process fidelity of the noisy operation, $\tilde{G}$, to the noiseless operation, G, [44, 48]. If the noisy operation $\tilde{G}$ is free of error, then the process fidelity will be 1. We repeat experiments with different input pure states. We generate the input states at random in the same manner as in Reference [12], by selecting random polar coordinates on the Bloch sphere for each qubit and report the average process fidelity.

Numerous quantum simulators exist (References [72, 77, 124, 142, 145]), many of which would be suitable to run our simulations. However, as we are implementing algorithms and incorporating noise models from a variety of sources, and did not want to unintentionally bias our experiments by relying on any specific software, we chose to run our simulations with the statistical programming language R [121]. This allows us to fully and independently define our experiments. Additionally, R is highly optimized to perform matrix multiplication, which is the essential component for density matrix simulation. Our source code is available in Reference [122], which is an extension of

Table 5. Noise Levels Tested for Each Noise Type

Error Level	Pauli	Coherent	Pauli+Coherent	Amplitude Damping
0	0	0π	0, 0π	0
1	0.01	$\pi/30$	0.01, $\pi/30$	0.01
2	0.02	$\pi/15$	0.02, $\pi/15$	0.02
3	0.03	$\pi/10$	0.03, $\pi/10$	0.03

Noise is inserted in every qubit in every cycle, regardless if it is being operated on. Pauli noise rate refers to probability of inserting and X, Y, or Z gate. Coherent noise is the rotation angle applied. For Pauli+Coherent, only X Pauli gates and X rotations are applied to align with the model in Reference [63]. Amplitude Damping error rate refers to the parameter γ [149], which is the probability of relaxation to the ground state.



(a) Pauli Noise (b) Coherent Noise (c) Pauli+Coherent Noise (d) Amplitude Damping

Fig. 2. Idle circuit with four qubits under various noise models.

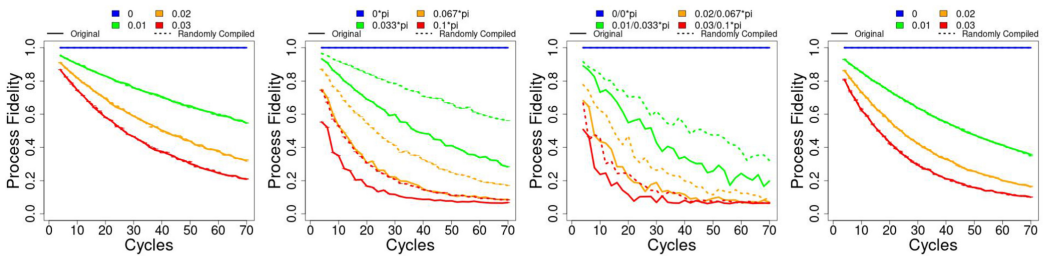
our R package QuantumOps [123]. To perform the gate decomposition, we rely on the gridsynth algorithm [131].

Performing simulation with density matrices allows us to avoid much Monte Carlo simulation. However, Monte Carlo simulation is still needed due to the randomness introduced by our random input states, use of RC, and the random circuits for the random circuit benchmark.

9 RESULTS

For the Idle and Random circuits we sweep the noise levels over the values listed in Table 5 and plot the effect on the fidelity for circuits of different lengths. As the Addition, QFT, and QAOA circuits have a constant depth, we sweep the four error models over a fine-grained range and plot the process fidelity or accuracy of the output versus the error rate.

Results for the Idle circuit are shown in Figure 2 and results for the Random circuit are shown in Figure 3. Both circuits are performed from 2 to 70 cycles. Cycles here indicate the length of a single gate. Note that error level 0 shows a process fidelity of 1, meaning there is no corruption of the quantum state. It is highly noticeable how coherent noise affects the Idle and Random circuits differently. For the Idle circuit, coherent noise has an immediate drastic impact on the fidelity. We must note the strange behavior of the Idle circuit under coherent noise. Due to our simplified coherent noise model, the fidelity returns with a periodicity determined by the constant angle of rotation. This is unlikely to be a physically realistic phenomenon, and even if it was, *it would not be possible to exploit this fact unless one was completely aware of the exact effects of the physical noise.* As physical noise is difficult to model and predict, it is highly



(a) Pauli Noise (b) Coherent Noise (c) Pauli+Coherent Noise (d) Amplitude Damping

Fig. 3. Random circuit with four qubits under various noise models.

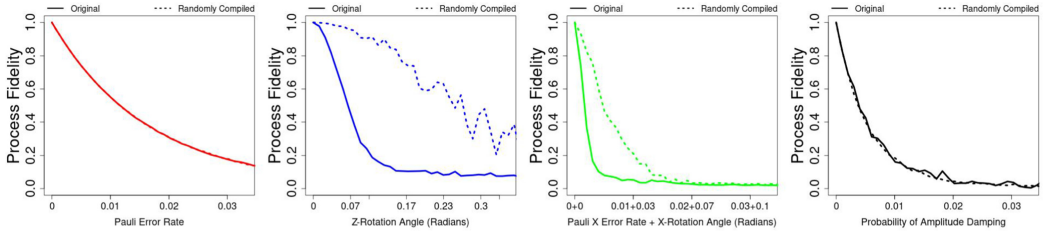
unlikely one would have such knowledge. Note that randomized compiling removes this periodic effect.

The true observation from the simulation of coherent noise on the Idle circuit is the immediate destructive nature of even a slight coherent noise source. Note that randomized compiling mitigates this impact and causes the coherent noise to have the same effect as stochastic Pauli noise. Interestingly, this same coherent noise is not as destructive on the Random circuit. In fact, even without Randomized Compiling, the error decays exponentially just as it does under stochastic Pauli noise. This finding is consistent with [18], which says that coherent noise will affect randomized circuits much differently than idle or cyclic circuits. The coherent noise gets "smeared out" by the randomization inherent in a random circuit. Additionally, this suggests that the randomized benchmark circuits in Quantum Volume [39], depending on the gate decomposition, may not be representative of many quantum circuits. Noteworthy is that Randomized Compiling may not be necessary if the circuit already contains a high degree of randomness. However, the vast majority of useful quantum algorithms do not have such structure.

The chosen quantum noise model has a drastic impact on the performance of quantum algorithms. Hence, one must be sure that the assumptions on the noise present in a physical system are appropriate. Additionally, the effect of the quantum noise is largely determined by the nature of the quantum algorithm being performed. Hence, one must be cautious when choosing quantum algorithms for benchmarks.

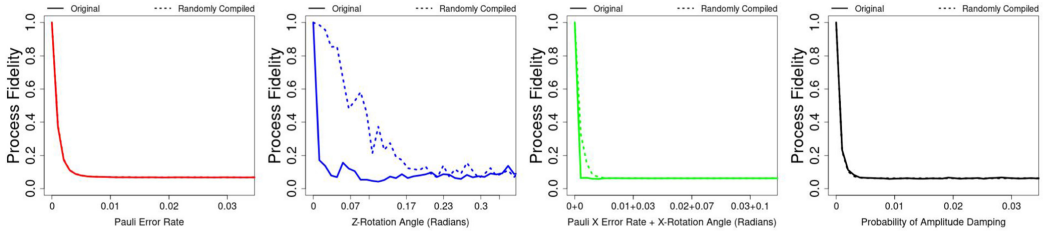
Results for the addition circuit are shown in Figure 4, the parameterized rotation gate version of QFT in Figure 5, and the Clifford+T version of QFT in Figure 6. Note that circuits using the parameterized rotation gates cannot be randomly compiled. Randomized Compiling produces a significant increase in fidelity when coherent noise is present.

Note that under Pauli noise or Amplitude Damping, Randomized Compiling does not provide a significant improvement. For Pauli noise, as it is already entirely random, it remains unmodified by the random compilation. This is noted by the inventors of RC [159]. As RC is designed to mitigate coherent rotations of the qubit state, it is also intuitive that it would not significantly mitigate Amplitude Damping, which models energy loss of the system. However, RC may help in some specific circumstances, such as when a qubit is held in the excited $|1\rangle$ for an extended period of time. As Amplitude Damping models the collapse from $|1\rangle$ to $|0\rangle$, the $|1\rangle$ state is more vulnerable. As noted by the authors of Reference [127], Amplitude Damping is more destructive if the quantum data in a program contains more qubits in the $|1\rangle$ state. RC could make the state oscillate, reducing the amount of time the qubit will spend in the excited, more vulnerable state. However, this condition was not present in our benchmarks.



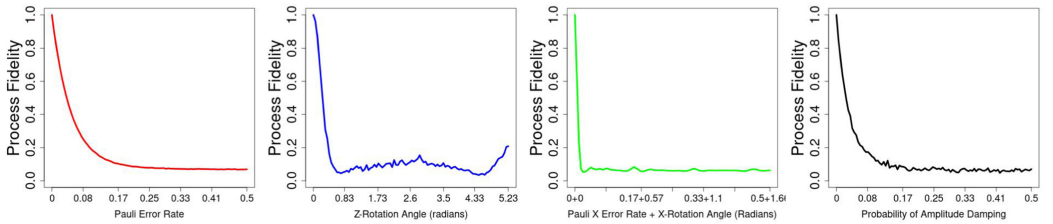
(a) Pauli Noise (b) Coherent Noise (c) Pauli+Coherent Noise (d) Amplitude Damping

Fig. 4. 2-bit addition circuit under various noise models.



(a) Pauli Noise (b) Coherent Noise (c) Pauli+Coherent Noise (d) Amplitude Damping

Fig. 5. 4-qubit QFT (Clifford+T) under various noise models.

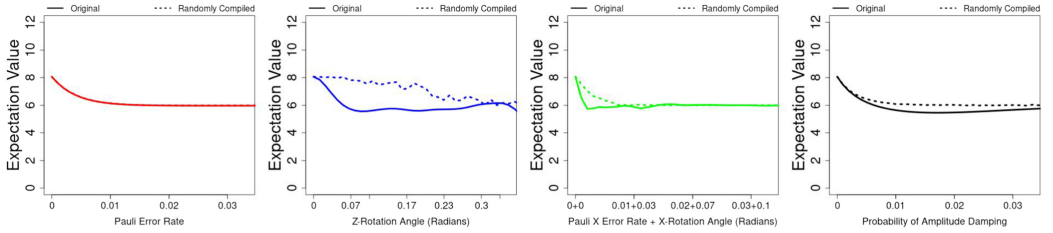


(a) Pauli Noise (b) Coherent Noise (c) Pauli+Coherent Noise (d) Amplitude Damping

Fig. 6. 4-qubit QFT with parameterized rotation gates under various noise models.

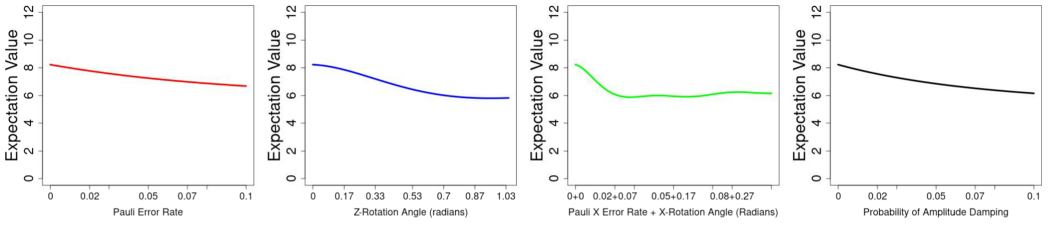
While a critical tool to enable scalable quantum computing, especially for modern machines, RC cannot be used as a generalized noise mitigation technique. As noted by the original paper [159], randomized compiling is used to "convert" coherent noise to stochastic Pauli noise. This is significant, as modern, physical quantum computers are dominated by coherent errors. Additionally, this conversion to Pauli noise maintains the validity of previously established proofs that have assumed Pauli noise.

Results for the Clifford+T QAOA are shown in Figure 7. If no error is present, then the QAOA circuit will produce an expectation value of 8.3, as this is 0.6924× of the maximum cut (12), which is expected for a QAOA circuit with $p = 1$. The expectation value of a random guess for the max-cut problem is at least 50% that of the maximum cut [65]. For the specific problem we used, a random guess produced an expectation value exactly 50% of that maximum. Hence, increasing the noise, which increases the entropy of the output, generally caused the expectation value to decay to 6. Noticeably, the expectation value drops to that of a random guess even for very low levels of noise. Amplitude Damping noise has significant potential to drop the expectation value below 50%, as



(a) Pauli Noise (b) Coherent Noise (c) Pauli+Coherent Noise (d) Amplitude Damping

Fig. 7. QAOA (Clifford+T) expectation value under different noise models.



(a) Pauli Noise (b) Coherent Noise (c) Pauli+Coherent Noise (d) Amplitude Damping

Fig. 8. QAOA with parameterized rotation gates expectation value under different noise models. As the circuit is significantly shorter than Clifford+T QAOA, the noise is tested over a significantly larger range.

high levels of this noise will cause the state to transition more toward the $|00\ldots0\rangle$ state. The $|00\ldots0\rangle$ state does not satisfy any of the input problem's constraints and hence produces an expectation value of 0. Notice that RC prevents this drop below 50%.

QAOA using parametrized rotation gates is shown in Figure 8. Due to not using decomposition, the circuit is much shorter and hence significantly less impacted by noise. Note that this gate set is not compatible with randomized compiling or quantum error correction. Hence, it is not scalable but still feasible for the small cases that we are considering. Despite the difference in gate sets, this version of QAOA shows similar expectation value patterns due to the different noise types. The expectation value tends to converge toward a value that is 50% of the maximum. Again, a combination of Pauli and coherent noise is particularly destructive.

10 CONCLUSION

Computer architecture uses layers of abstraction to manage complex problems. This approach has already been applied to quantum computing. Unfortunately, quantum systems are notorious at defying abstraction and simplifying assumptions. It is easy to make invalid assumptions and generate inaccurate results. Here, we showed that quantum noise is more complex and difficult to model than is often assumed. This has profound effects everywhere, and can be felt significantly even at higher levels of the system stack. This complicates the task of benchmarking, which is already challenging and full of subtlety for classical computers. The noise model, the target application, and the performance metric all need to be carefully considered.

REFERENCES

[1] Scott Aaronson. 2019. Shadow tomography of quantum states. *SIAM J. Comput.* 49, 5 (2019), STOC18–368.
[2] Scott Aaronson. 2020. Turn down the quantum volume. Retrieved from https://www.scottaaronson.com/blog/?p=4649.

[3] Scott Aaronson and Lijie Chen. 2016. Complexity-theoretic foundations of quantum supremacy experiments. arXiv:1612.05903. Retrieved from https://arxiv.org/abs/1612.05903.
[4] Scott Aaronson and Daniel Gottesman. 2004. Improved simulation of stabilizer circuits. *Phys. Rev. A* 70, 5 (2004), 052328.
[5] Gadi Aleksandrowicz, Thomas Alexander, Panagiotis Barkoutsos, Luciano Bello, Yael Ben-Haim, David Bucher, Francisco Jose Cabrera-Hernández, Jorge Carballo-Franquis, Adrian Chen, Chun-Fu Chen, Jerry M. Chow, Antonio D. Córcoles-Gonzales, Abigail J. Cross, Andrew Cross, Juan Cruz-Benito, Chris Culver, Salvador De La Puente González, Enrique De La Torre, Delton Ding, Eugene Dumitrescu, Ivan Duran, Pieter Eendebak, Mark Everitt, Ismael Faro Sertage, Albert Frisch, Andreas Fuhrer, Jay Gambetta, Borja Godoy Gago, Juan Gomez-Mosquera, Donny Greenberg, Ikko Hamamura, Vojtech Havlicek, Joe Hellmers, łukasz Herok, Hiroshi Horii, Shaohan Hu, Takashi Imamichi, Toshinari Itoko, Ali Javadi-Abhari, Naoki Kanazawa, Anton Karazeev, Kevin Krsulich, Peng Liu, Yang Luh, Yunho Maeng, Manoel Marques, Francisco Jose Martín-Fernández, Douglas T. McClure, David McKay, Srujan Meesala, Antonio Mezzacapo, Nikolaj Moll, Diego Moreda Rodríguez, Giacomo Nannicini, Paul Nation, Pauline Ollitrault, Lee James O'Riordan, Hanhee Paik, Jesús Pérez, Anna Phan, Marco Pistoia, Viktor Prutyanov, Max Reuter, Julia Rice, Abdón Rodríguez Davila, Raymond Harry Putra Rudy, Mingi Ryu, Ninad Sathaye, Chris Schnabel, Eddie Schoute, Kanav Setia, Yunong Shi, Adenilton Silva, Yukio Siraichi, Seyon Sivarajah, John A. Smolin, Mathias Soeken, Hitomi Takahashi, Ivano Tavernelli, Charles Taylor, Pete Taylour, Kenso Trabing, Matthew Treinish, Wes Turner, Desiree Vogt-Lee, Christophe Vuillot, Jonathan A. Wildstrom, Jessica Wilson, Erick Winston, Christopher Wood, Stephen Wood, Stefan Wörner, Ismail Yunus Akhalwaya, and Christa Zoufal. 2019. Qiskit: An open-source framework for quantum computing.
[6] Dorit Aharonov and Michael Ben-Or. 2008. Fault-tolerant quantum computation with constant error rate. *SIAM J. Comput.* (2008).
[7] Rafael N. Alexander, Peter S. Turner, and Stephen D. Bartlett. 2016. Randomized benchmarking in measurement-based quantum computing. *Phys. Rev. A* 94, 3 (2016), 032303.
[8] Matthew Amy, Dmitri Maslov, Michele Mosca, and Martin Roetteler. 2013. A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 32, 6 (2013), 818–830.
[9] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, et al. 2019. Quantum supremacy using a programmable superconducting processor. *Nature* 574, 7779 (2019), 505–510.
[10] Eyal Bairey, Itai Arad, and Netanel H. Lindner. 2019. Learning a local Hamiltonian from local measurements. *Phys. Rev. Lett.* 122, 2 (2019), 020504.
[11] Harrison Ball, Thomas M. Stace, Steven T. Flammia, and Michael J. Biercuk. 2016. Effect of noise correlations on randomized benchmarking. *Phys. Rev. A* 93, 2 (2016), 022303.
[12] Jeff P. Barnes, Colin J. Trout, Dennis Lucarelli, and B. D. Clader. 2017. Quantum error-correction failure distributions: Comparison of coherent and stochastic error models. *Phys. Rev. A* 95, 6 (2017), 062338.
[13] Stefanie J. Beale, Joel J. Wallman, Mauricio Gutiérrez, Kenneth R. Brown, and Raymond Laflamme. 2018. Quantum error correction decoheres noise. *Phys. Rev. Lett.* 121, 19 (2018), 190501.
[14] Ethan Bernstein and Umesh Vazirani. 1997. Quantum complexity theory. *SIAM J. Comput.* 26, 5 (1997), 1411–1473.
[15] Christian Bienia, Sanjeev Kumar, Jaswinder Pal Singh, and Kai Li. 2008. The PARSEC benchmark suite: Characterization and architectural implications. In *Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques*. ACM, 72–81.
[16] Lev S. Bishop, Sergey Bravyi, Andrew Cross, Jay M. Gambetta, and John Smolin. 2017. Quantum Volume. Technical Report.
[17] Robin Blume-Kohout, John King Gamble, Erik Nielsen, Kenneth Rudinger, Jonathan Mizrahi, Kevin Fortier, and Peter Maunz. 2017. Demonstration of qubit operations below a rigorous fault tolerance threshold with gate set tomography. *Nat. Commun.* 8 (2017), 14485.
[18] Robin Blume-Kohout and Kevin C. Young. 2019. A volumetric framework for quantum computer benchmarks. arXiv:1904.05546. Retrieved from https://arxiv.org/abs/1904.05546.
[19] Robin J. Blume-Kohout. 2017. *How Distinguishable Are Two Quantum Processes or What Is the Error Rate of a Quantum Gate?* Technical Report. Sandia National Lab, Albuquerque, NM.
[20] Robin J. Blume-Kohout and Kevin Young. 2019. *Metrics and Benchmarks for Quantum Processors: State of Play.* Technical Report. Sandia National Lab, Albuquerque, NM.
[21] Sergio Boixo, Sergei V. Isakov, Vadim N. Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J. Bremner, John M. Martinis, and Hartmut Neven. 2018. Characterizing quantum supremacy in near-term devices. *Nat. Phys.* 14, 6 (2018), 595–600.
[22] Kristine Boone, Arnaud Carignan-Dugas, Joel J. Wallman, and Joseph Emerson. 2019. Randomized benchmarking under different gate sets. *Phys. Rev. A* 99, 3 (2019), 032329.

[23] Adam Bouland, Bill Fefferman, Chinmay Nirkhe, and Umesh Vazirani. 2019. On the complexity and verification of quantum random circuit sampling. *Nat. Phys.* 15, 2 (2019), 159–163.
[24] Mark D. Bowdrey, Daniel K. L. Oi, Anthony J. Short, Konrad Banaszek, and Jonathan A. Jones. 2002. Fidelity of single qubit maps. *Phys. Lett. A* 294, 5–6 (2002), 258–260.
[25] Stephen Boyd, Stephen P. Boyd, and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.
[26] Sergey Bravyi, Matthias Englbrecht, Robert König, and Nolan Peard. 2018. Correcting coherent errors with surface codes. *npj Quant. Inf.* 4, 1 (2018), 55.
[27] Teresa Brecht, Wolfgang Pfaff, Chen Wang, Yiwen Chu, Luigi Frunzio, Michel H. Devoret, and Robert J. Schoelkopf. 2016. Multilayer microwave integrated quantum circuits for scalable quantum computing. *npj Quant. Inf.* 2 (2016), 16002.
[28] Michael J. Bremner, Ashley Montanaro, and Dan J. Shepherd. 2016. Average-case complexity versus approximate simulation of commuting quantum computations. *Phys. Rev. Lett.* 117, 8 (2016), 080501.
[29] Joseph W. Britton, Brian C. Sawyer, Adam C. Keith, C.-C. Joseph Wang, James K. Freericks, Hermann Uys, Michael J. Biercuk, and John J. Bollinger. 2012. Engineered two-dimensional Ising interactions in a trapped-ion quantum simulator with hundreds of spins. *Nature* 484, 7395 (2012), 489–492.
[30] Colin D. Bruzewicz, John Chiaverini, Robert McConnell, and Jeremy M. Sage. 2019. Trapped-ion quantum computing: Progress and challenges. *Appl. Phys. Rev.* 6, 2 (2019), 021314.
[31] R. Cabrera and W. E. Baylis. 2007. Average fidelity in n-qubit systems. *Phys. Lett. A* 368, 1–2 (2007), 25–28.
[32] Yudong Cao, Jonathan Romero, Jonathan P. Olson, Matthias Degroote, Peter D. Johnson, Mária Kieferová, Ian D. Kivlichan, Tim Menke, Borja Peropadre, Nicolas P. D. Sawaya, et al. 2018. Quantum chemistry in the age of quantum computing. arXiv:1812.09976. Retrieved from https://arxiv.org/abs/1812.09976.
[33] Arnaud Carignan-Dugas, Joel J. Wallman, and Joseph Emerson. 2015. Characterizing universal gate sets via dihedral benchmarking. *Phys. Rev. A* 92, 6 (2015), 060302.
[34] Christopher Chamberland, Joel Wallman, Stefanie Beale, and Raymond Laflamme. 2017. Hard decoding algorithm for optimizing thresholds under general markovian noise. *Phys. Rev. A* 95, 4 (2017), 042332.
[35] Kai-Wen Cheng and Chien-Cheng Tseng. 2002. Quantum full adder and subtractor. *Electr. Lett.* 38, 22 (2002), 1343–1344.
[36] Jerry M. Chow, Jay M. Gambetta, Antonio D. Corcoles, Seth T. Merkel, John A. Smolin, Chad Rigetti, S. Poletto, George A. Keefe, Mary B. Rothwell, John R. Rozen, et al. 2012. Universal quantum gate set approaching fault-tolerant thresholds with superconducting qubits. *Phys. Rev. Lett.* 109, 6 (2012), 060501.
[37] Isaac L. Chuang and Michael A. Nielsen. 1997. Prescription for experimental determination of the dynamics of a quantum black box. *J. Mod. Opt.* 44, 11–12 (1997), 2455–2467.
[38] Marcus Cramer, Martin B. Plenio, Steven T. Flammia, Rolando Somma, David Gross, Stephen D. Bartlett, Olivier Landon-Cardinal, David Poulin, and Yi-Kai Liu. 2010. Efficient quantum state tomography. *Nat. Commun.* 1 (2010), 149.
[39] Andrew W. Cross, Lev S. Bishop, Sarah Sheldon, Paul D. Nation, and Jay M. Gambetta. 2019. Validating quantum computers using randomized model circuits. *Phys. Rev. A* 100, 3 (2019), 032328.
[40] Christoph Dankert, Richard Cleve, Joseph Emerson, and Etera Livine. 2009. Exact and approximate unitary 2-designs and their application to fidelity estimation. *Phys. Rev. A* 80, 1 (2009), 012304.
[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR'09)*.
[42] Casey Duckering, Jonathan M. Baker, David I. Schuster, and Frederic T. Chong. 2020. Virtualized logical qubits: A 2.5 D architecture for error-corrected quantum computing. In *Proceedings of the 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'20)*. IEEE, 173–185.
[43] Joseph Emerson, Robert Alicki, and Karol Życzkowski. 2005. Scalable noise estimation with random unitary operators. *J. Opt. B: Quant. Semiclass. Opt.* 7, 10 (2005), S347.
[44] Alexander Erhard, Joel James Wallman, Lukas Postler, Michael Meth, Roman Stricker, Esteban Adrian Martinez, Philipp Schindler, Thomas Monz, Joseph Emerson, and Rainer Blatt. 2019. Characterizing large-scale quantum computers via cycle benchmarking. arXiv:1902.08543. Retrieved from https://arxiv.org/abs/1902.08543.
[45] Tim J. Evans, Robin Harper, and Steven T. Flammia. 2019. Scalable bayesian hamiltonian learning. arXiv:1912.07636. Retrieved from https://arxiv.org/abs/1912.07636.
[46] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. 2014. A quantum approximate optimization algorithm. arXiv:1411.4028. Retrieved from https://arxiv.org/abs/1411.4028.
[47] Samuele Ferracin, Theodoros Kapourniotis, and Animesh Datta. 2019. Accrediting outputs of noisy intermediate-scale quantum computing devices. *New J. Phys.* 21, 11 (2019), 113038.
[48] Steven T. Flammia and Yi-Kai Liu. 2011. Direct fidelity estimation from few Pauli measurements. *Phys. Rev. Lett.* 106, 23 (2011), 230501.
[49] Austin G. Fowler. 2013. Coping with qubit leakage in topological codes. *Phys. Rev. A* 88, 4 (2013), 042308.

[50] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. 2012. Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A* 86, 3 (2012), 032324.
[51] Christopher A. Fuchs and Jeroen Van De Graaf. 1999. Cryptographic distinguishability measures for quantum-mechanical states. *IEEE Trans. Inf. Theory* 45, 4 (1999), 1216–1227.
[52] Jay M. Gambetta, A. D. Córcoles, Seth T. Merkel, Blake R. Johnson, John A. Smolin, Jerry M. Chow, Colm A. Ryan, Chad Rigetti, S. Poletto, Thomas A. Ohki, et al. 2012. Characterization of addressability by simultaneous randomized benchmarking. *Phys. Rev. Lett.* 109, 24 (2012), 240504.
[53] Michael R. Geller and Zhongyuan Zhou. 2013. Efficient error models for fault-tolerant architectures and the Pauli twirling approximation. *Phys. Rev. A* 88, 1 (2013), 012314.
[54] Joydip Ghosh, Austin G. Fowler, and Michael R. Geller. 2012. Surface code with decoherence: An analysis of three superconducting architectures. *Phys. Rev. A* 86, 6 (2012), 062318.
[55] Alexei Gilchrist, Nathan K. Langford, and Michael A. Nielsen. 2005. Distance measures to compare real and ideal quantum processes. *Phys. Rev. A* 71, 6 (2005), 062310.
[56] Michel X. Goemans and David P. Williamson. 1995. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM* 42, 6 (1995), 1115–1145.
[57] Christian Gogolin, Martin Kliesch, Leandro Aolita, and Jens Eisert. 2013. Boson-sampling in the light of sample complexity. arXiv:1306.3995. Retrieved from https://arxiv.org/abs/1306.3995.
[58] Daniel Gottesman. 1997. Stabilizer codes and quantum error correction. arXiv:quant-ph/9705052. Retrieved from https://arxiv.org/abs/9705052.
[59] Daniel Gottesman. 1998. The Heisenberg representation of quantum computers. arXiv:quant-ph/9807006. Retrieved from https://arxiv.org/abs/9807006.
[60] Daniel Gottesman. 2010. An introduction to quantum error correction and fault-tolerant quantum computation. In *Quantum Information Science and Its Contributions to Mathematics*, Proceedings of Symposia in Applied Mathematics, Vol. 68. 13–58.
[61] Christopher E. Granade, Christopher Ferrie, Nathan Wiebe, and David G. Cory. 2012. Robust online Hamiltonian learning. *New J. Phys.* 14, 10 (2012), 103013.
[62] Daniel Greenbaum. 2015. Introduction to quantum gate set tomography. arXiv:1509.02921. Retrieved from https://arxiv.org/abs/1509.02921.
[63] Daniel Greenbaum and Zachary Dutton. 2017. Modeling coherent errors in quantum error correction. *Quant. Sci. Technol.* 3, 1 (2017), 015007.
[64] Robert B. Griffiths. 2003. *Consistent Quantum Theory*. Cambridge University Press.
[65] Grove Documentation 2019. Retrieved March 6, 2012 from https://grove-docs.readthedocs.io/en/latest/vqe.html.
[66] Lov K. Grover. 1996. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*. ACM, 212–219.
[67] Gian Giacomo Guerreschi and Anne Y. Matsuura. 2019. QAOA for Max-Cut requires hundreds of qubits for quantum speed-up. *Sci. Rep.* 9 (2019).
[68] John L. Gustafson and Quinn O. Snell. 1995. HINT: A new way to measure computer performance. In *Proceedings of the 28th Annual Hawaii International Conference on System Sciences*, Vol. 2. IEEE, 392–401.
[69] Mauricio Gutiérrez and Kenneth R. Brown. 2015. Comparison of a quantum error-correction threshold for exact and approximate errors. *Phys. Rev. A* 91, 2 (2015), 022335.
[70] Mauricio Gutiérrez, Conor Smith, Livia Lulushi, Smitha Janardan, and Kenneth R. Brown. 2016. Errors and pseudothresholds for incoherent and coherent noise. *Phys. Rev. A* 94, 4 (2016), 042338.
[71] Mauricio Gutiérrez, Lukas Svec, Alexander Vargo, and Kenneth R. Brown. 2013. Approximation of realistic errors by Clifford channels and Pauli measurements. *Phys. Rev. A* 87, 3 (2013), 030302.
[72] Thomas Häner, Damian S. Steiger, Mikhail Smelyanskiy, and Matthias Troyer. 2016. High performance emulation of quantum circuits. In *SC'16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 866–874.
[73] Philipp Hauke, Fernando M. Cucchietti, Luca Tagliacozzo, Ivan Deutsch, and Maciej Lewenstein. 2012. Can one trust quantum simulators? *Rep. Progr. Phys.* 75, 8 (2012), 082401.
[74] Jonas Helsen, Xiao Xue, Lieven M. K. Vandersypen, and Stephanie Wehner. 2019. A new class of efficient randomized benchmarking protocols. *npj Quant. Inf.* 5, 1 (2019), 1–9.
[75] John L. Hennessy and David A. Patterson. 2011. *Computer Architecture: A Quantitative Approach*. Elsevier.
[76] Hsin-Yuan Huang, Richard Kueng, and John Preskill. 2020. Predicting many properties of a quantum system from very few measurements. arXiv:2002.08953. Retrieved from https://arxiv.org/abs/2002.08953.
[77] Ali JavadiAbhari, Shruti Patil, Daniel Kudrow, Jeff Heckey, Alexey Lvov, Frederic T. Chong, and Margaret Martonosi. 2014. ScaffCC: A framework for compilation and analysis of quantum computing programs. In *Proceedings of the 11th ACM Conference on Computing Frontiers*. ACM, 1.

[78] Gil Kalai and Guy Kindler. 2014. Gaussian noise sensitivity and BosonSampling. arXiv:1409.3093. Retrieved from https://arxiv.org/abs/1409.3093.
[79] Nader Khammassi, Imran Ashraf, Xiang Fu, Carmen G. Almudever, and Koen Bertels. 2017. QX: A high-performance quantum computer simulation platform. In *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE'17)*. IEEE, 464–469.
[80] Kihwan Kim, M.-S. Chang, Simcha Korenblit, Rajibul Islam, Emily E. Edwards, James K. Freericks, G.-D. Lin, L.-M. Duan, and Christopher Monroe. 2010. Quantum simulation of frustrated Ising spins with trapped ions. *Nature* 465, 7298 (2010), 590–593.
[81] Shelby Kimmel, Marcus P. da Silva, Colm A. Ryan, Blake R. Johnson, and Thomas Ohki. 2014. Robust extraction of tomographic information via randomized benchmarking. *Phys. Rev. X* 4, 1 (2014), 011050.
[82] Vadym Kliuchnikov, Dmitri Maslov, and Michele Mosca. 2012. Fast and efficient exact synthesis of single qubit unitaries generated by Clifford and T gates. arXiv:1206.5236. Retrieved from https://arxiv.org/abs/1206.5236.
[83] Emanuel Knill, Raymond Laflamme, and Wojciech H. Zurek. 1998. Resilient quantum computation: Error models and thresholds. *Proc. Roy. Soc. Lond. Ser. A* 454, 1969 (1998), 365–384.
[84] Emanuel Knill, Dietrich Leibfried, Rolf Reichle, Joe Britton, R. Brad Blakestad, John D. Jost, Chris Langer, Roee Ozeri, Signe Seidelin, and David J. Wineland. 2008. Randomized benchmarking of quantum gates. *Phys. Rev. A* 77, 1 (2008), 012307.
[85] Richard Kueng, David M. Long, Andrew C. Doherty, and Steven T. Flammia. 2016. Comparing experiments to the fault-tolerance threshold. *Phys. Rev. Lett.* 117, 17 (2016), 170502.
[86] Adrian Kügel. 2010. Improved Exact Solver for the Weighted MAX-SAT Problem. *Pos@ sat* 8 (2010), 15–27.
[87] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.
[88] Gushu Li, Yufei Ding, and Yuan Xie. 2019. SANQ: A simulation framework for architecting noisy intermediate-scale quantum computing system. arXiv:1904.11590. Retrieved from https://arxiv.org/abs/1904.11590.
[89] Gushu Li, Yufei Ding, and Yuan Xie. 2019. Tackling the qubit mapping problem for NISQ-era quantum devices. In *Proceedings of the 24th International Conference on Architectural Support for Programming Languages and Operating Systems*. 1001–1014.
[90] Muyuan Li, Mauricio Gutiérrez, Stanley E. David, Alonzo Hernandez, and Kenneth R. Brown. 2017. Fault tolerance with bare ancillary qubits for a [[7, 1, 3]] code. *Phys. Rev. A* 96, 3 (2017), 032341.
[91] David J. Lilja. 2005. *Measuring Computer Performance: A Practitioner's Guide*. Cambridge University Press.
[92] Norbert M. Linke, Dmitri Maslov, Martin Roetteler, Shantanu Debnath, Caroline Figgatt, Kevin A. Landsman, Kenneth Wright, and Christopher Monroe. 2017. Experimental comparison of two quantum computing architectures. *Proc. Natl. Acad. Sci. U.S.A.* 114, 13 (2017), 3305–3310.
[93] Seth Lloyd. 2013. Pure state quantum statistical mechanics and black holes. arXiv:1307.0378. Retrieved from https://arxiv.org/abs/1307.0378.
[94] Michael Loceff. 2015. A Course in Quantum Computing (for the Community College). Foothill College. Retrieved from https://lapastillaroja.net/wp-content/uploads/2016/09/Intro_to_QC_Vol_1_Loceff.pdf.
[95] Shunlong Luo and Qiang Zhang. 2004. Informational distance on quantum-state space. *Physical Review A* 69, 3 (2004), 032106.
[96] E. Magesan, J. M. Gambetta, and J. Emerson. [n.d.]. Robust randomized benchmarking of quantum processes. arXiv:1009.3639. Retrieved from https://arxiv.org/abs/1009.3639.
[97] Easwar Magesan, Jay M. Gambetta, and Joseph Emerson. 2011. Scalable and robust randomized benchmarking of quantum processes. *Phys. Rev. Lett.* 106, 18 (2011), 180504.
[98] Easwar Magesan, Jay M. Gambetta, and Joseph Emerson. 2012. Characterizing quantum gates via randomized benchmarking. *Phys. Rev. A* 85, 4 (2012), 042311.
[99] Easwar Magesan, Jay M. Gambetta, Blake R. Johnson, Colm A. Ryan, Jerry M. Chow, Seth T. Merkel, Marcus P. Da Silva, George A. Keefe, Mary B. Rothwell, Thomas A. Ohki, et al. 2012. Efficient measurement of quantum gate error by interleaved randomized benchmarking. *Phys. Rev. Lett.* 109, 8 (2012), 080505.
[100] Easwar Magesan, Daniel Puzzuoli, Christopher E. Granade, and David G. Cory. 2013. Modeling quantum noise for efficient testing of fault-tolerant circuits. *Phys. Rev. A* 87, 1 (2013), 012324.
[101] Margaret Martonosi and Martin Roetteler. 2019. Next steps in quantum computing: Computer science's role. arXiv:1903.10541. Retrieved from https://arxiv.org/abs/1903.10541.
[102] Sam McArdle, Suguru Endo, Alan Aspuru-Guzik, Simon Benjamin, and Xiao Yuan. 2018. Quantum computational chemistry. arXiv:1808.10402. Retrieved from https://arxiv.org/abs/1808.10402.
[103] Seth T. Merkel, Jay M. Gambetta, John A. Smolin, Stefano Poletto, Antonio D. Córcoles, Blake R. Johnson, Colm A. Ryan, and Matthias Steffen. 2013. Self-consistent quantum process tomography. *Phys. Rev. A* 87, 6 (2013), 062119.

[104] Seth T. Merkel, Emily J. Pritchett, and Bryan H. Fong. 2018. Randomized benchmarking as convolution: Fourier analysis of gate dependent errors. arXiv:1804.05951. Retrieved from https://arxiv.org/abs/1804.05951.
[105] Tzvetan S. Metodi and Frederic T. Chong. 2006. Quantum computing for computer architects. *Synth. Lect. Comput. Arch.* 1, 1 (2006), 1–154.
[106] Naomi H. Nickerson and Benjamin J. Brown. 2019. Analysing correlated noise on the surface code using adaptive decoding algorithms. *Quantum* 3 (2019), 131.
[107] Erik Nielsen, John King Gamble, Kenneth Rudinger, Travis Scholten, Kevin Young, and Robin Blume-Kohout. 2020. Gate set tomography. arXiv:2009.07301. Retrieved from https://arxiv.org/abs/2009.07301.
[108] Erik Nielsen, Kenneth Rudinger, Timothy Proctor, Antonio Russo, Kevin Young, and Robin Blume-Kohout. 2020. Probing quantum processor performance with pyGSTi. arXiv:2002.12476. Retrieved from https://arxiv.org/abs/2002.12476.
[109] Michael A. Nielsen and Isaac Chuang. 2002. Quantum computation and quantum information. 558–559.
[110] Jonathan Olson, Yudong Cao, Jonathan Romero, Peter Johnson, Pierre-Luc Dallaire-Demers, Nicolas Sawaya, Prineha Narang, Ian Kivlichan, Michael Wasielewski, and Alán Aspuru-Guzik. 2017. Quantum information and computation for chemistry. arXiv:1706.05413. Retrieved from https://arxiv.org/abs/1706.05413.
[111] Mark Oskin, Frederic T. Chong, and Isaac L. Chuang. 2002. A practical architecture for reliable quantum computers. *Computer* 1 (2002), 79–87.
[112] Matteo Paris and Jaroslav Rehacek. 2004. *Quantum State Estimation*. Vol. 649. Springer Science & Business Media.
[113] Pennylane: QAOA for MaxCut 2019. Retrieved September 26, 2020 from https://pennylane.ai/qml/demos/tutorial_qaoa_maxcut.html.
[114] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'brien. 2014. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* 5 (2014), 4213.
[115] Sandu Popescu, Anthony J. Short, and Andreas Winter. 2006. Entanglement and the foundations of statistical mechanics. *Nat. Phys.* 2, 11 (2006), 754–758.
[116] J. F. Poyatos, J. Ignacio Cirac, and Peter Zoller. 1997. Complete characterization of a quantum process: The two-bit quantum gate. *Phys. Rev. Lett.* 78, 2 (1997), 390.
[117] J. S. Pratt and J. H. Eberly. 2001. Qubit cross talk and entanglement decay. *Phys. Rev. B* 64, 19 (2001), 195314.
[118] John Preskill. 1998. Fault-tolerant quantum computation. In *Introduction to Quantum Computation and Information*. World Scientific, 213–269.
[119] Timothy Proctor, Kenneth Rudinger, Kevin Young, Mohan Sarovar, and Robin Blume-Kohout. 2017. What randomized benchmarking actually measures. *Phys. Rev. Lett.* 119, 13 (2017), 130502.
[120] Qiskit: Randomized Benchmarking 2019. Retrieved September 29, 2020 from https://qiskit.org/textbook/ch-quantum-hardware/randomized-benchmarking.html.
[121] R Core Team. 2016. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. Retrieved from https://www.R-project.org/.
[122] Salonik Resch. [n.d.]. Quantum Noise Profiling. Retrieved from https://github.com/SalonikResch/QuantumNoiseProfiling.
[123] Salonik Resch. 2020. QuantumOps: Performs Common Linear Algebra Operations Used in Quantum Computing and Implements Quantum Algorithms. Retrieved from https://CRAN.R-project.org/package=QuantumOps. R package version 3.0.1.
[124] Martin Roetteler, Krysta M. Svore, Dave Wecker, and Nathan Wiebe. 2017. Design automation for quantum architectures. In *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE'17)*. IEEE, 1312–1317.
[125] Martin Rötteler. 2010. Quantum algorithms for highly non-linear Boolean functions. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 448–457.
[126] Mary Beth Ruskai. 1994. Beyond strong subadditivity? Improved bounds on the contraction of generalized relative entropy. *Rev. Math. Phys.* 6, 05a (1994), 1147–1161.
[127] Abdullah Ash Saki, Mahabubul Alam, and Swaroop Ghosh. 2019. Study of decoherence in quantum computers: A circuit-design perspective. arXiv:1904.04323. Retrieved from https://arxiv.org/abs/1904.04323.
[128] Mohan Sarovar, Timothy Proctor, Kenneth Rudinger, Kevin Young, Erik Nielsen, and Robin Blume-Kohout. 2019. Detecting crosstalk errors in quantum information processors. arXiv:1908.09855. Retrieved from https://arxiv.org/abs/1908.09855.
[129] Pradeep Kiran Sarvepalli, Andreas Klappenecker, and Martin Rötteler. 2009. Asymmetric quantum codes: Constructions, bounds and performance. *Proc. Roy. Soc. A* 465, 2105 (2009), 1645–1672.
[130] Maria Schuld, Alex Bocharov, Krysta Svore, and Nathan Wiebe. 2018. Circuit-centric quantum classifiers. arXiv:1804.00633. Retrieved from https://arxiv.org/abs/1804.00633.

[131] Peter Selinger. [n.d.]. Newsynth: Exact and approximate synthesis of quantum circuits. Retrieved from http://www.mathstat.dal.ca/~selinger/newsynth/.
[132] Peter Selinger. 2012. Efficient Clifford+ T approximation of single-qubit operators. arXiv:1212.6253. Retrieved from https://arxiv.org/abs/1212.6253.
[133] A. Shabani, R. L. Kosut, M. Mohseni, H. Rabitz, M. A. Broome, M. P. Almeida, A. Fedrizzi, and A. G. White. 2011. Efficient measurement of quantum dynamics via compressive sensing. *Phys. Rev. Lett.* 106, 10 (2011), 100401.
[134] Claude Elwood Shannon. 2001. A mathematical theory of communication. *ACM SIGMOBILE Mobile Comput. Commun. Rev.* 5, 1 (2001), 3–55.
[135] Peter W. Shor. 1995. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A* 52, 4 (1995), R2493.
[136] Peter W. Shor. 1999. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.* 41, 2 (1999), 303–332.
[137] Marcus Silva, Easwar Magesan, David W. Kribs, and Joseph Emerson. 2008. Scalable protocol for identification of correctable codes. *Phys. Rev. A* 78, 1 (2008), 012347.
[138] Jonathan Simon, Waseem S. Bakr, Ruichao Ma, M. Eric Tai, Philipp M. Preiss, and Markus Greiner. 2011. Quantum simulation of antiferromagnetic spin chains in an optical lattice. *Nature* 472, 7343 (2011), 307–312.
[139] James C. Spall et al. 1992. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans. Autom. Contr.* 37, 3 (1992), 332–341.
[140] D. Spehner, F. Illuminati, M. Orszag, and W. Roga. 2017. Geometric measures of quantum correlations with Bures and Hellinger distances. In *Lectures on General Quantum Correlations and Their Applications*. Springer, 105–157.
[141] Cloyce D. Spradling. 2007. SPEC CPU2006 benchmark tools. *ACM SIGARCH Comput. Arch. News* 35, 1 (2007), 130–134.
[142] Damian S. Steiger, Thomas Häner, and Matthias Troyer. 2018. ProjectQ: An open source software framework for quantum computing. *Quantum* 2, 49 (2018), 10–22331.
[143] Russell Stutz. 2020. Trapped ion quantum computing at Honeywell. *Bull. Am. Phys. Soc.* 65 (2020).
[144] Kevin J. Sung, Jiahao Yao, Matthew P. Harrigan, Nicholas C. Rubin, Zhang Jiang, Lin Lin, Ryan Babbush, and Jarrod R. McClean. 2020. Using models to improve optimizers for variational quantum algorithms. *Quant. Sci. Technol.* 5, 4 (2020), 044008.
[145] Krysta M. Svore, Alan Geller, Matthias Troyer, John Azariah, Christopher Granade, Bettina Heim, Vadym Kliuchnikov, Mariia Mykhailova, Andres Paz, and Martin Roetteler. 2018. Q#: Enabling scalable quantum computing and development with a high-level domain-specific language. arXiv:1803.00652. Retrieved from https://arxiv.org/abs/1803.00652.
[146] Krysta M. Svore and Matthias Troyer. 2016. The quantum future of computation. *Computer* 49, 9 (2016), 21–30.
[147] Swamit S. Tannu, Zachary A. Myers, Prashant J. Nair, Douglas M. Carmean, and Moinuddin K. Qureshi. 2017. Taming the instruction bandwidth of quantum computers via hardware-managed error correction. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*. 679–691.
[148] Swamit S. Tannu and Moinuddin K. Qureshi. 2019. Not all qubits are created equal: A case for variability-aware policies for NISQ-era quantum computers. In *Proceedings of the 24th International Conference on Architectural Support for Programming Languages and Operating Systems*. 987–999.
[149] Yu Tomita and Krysta M. Svore. 2014. Low-distance surface codes under realistic quantum noise. *Phys. Rev. A* 90, 6 (2014), 062320.
[150] Cozmin Ududec, Nathan Wiebe, and Joseph Emerson. 2013. Information-theoretic equilibration: The appearance of irreversibility under complex quantum dynamics. *Phys. Rev. Lett.* 111, 8 (2013), 080403.
[151] Austin P. Lund, J. Bremner Michael, and Ralph C. Timothy. 2017. Quantum sampling problems, Boson Sampling and quantum supremacy. *npj Quantum Information* 3, 1 (2017), 1–8.
[152] Wim Van Dam, Sean Hallgren, and Lawrence Ip. 2006. Quantum algorithms for some hidden shift problems. *SIAM J. Comput.* 36, 3 (2006), 763–778.
[153] K. Vogel and H. Risken. 1989. Determination of quasiprobability distributions in terms of probability distributions for the rotated quadrature phase. *Phys. Rev. A* 40, 5 (1989), 2847.
[154] J. J. Wallman. 2015. Error rates in quantum circuits. arXiv:1511.00727. Retrieved from https://arxiv.org/abs/1511.00727.
[155] Joel Wallman and Joseph Emerson. [n.d.]. Retrieved from https://www.quantumresource.org/pdfs/wallman.pptx.
[156] Joel Wallman, Chris Granade, Robin Harper, and Steven T. Flammia. 2015. Estimating the coherence of noise. *New J. Phys.* 17, 11 (2015), 113020.
[157] Joel J. Wallman. 2015. Bounding experimental quantum error rates relative to fault-tolerant thresholds. arXiv:1511.00727. Retrieved from https://arxiv.org/abs/1511.00727.
[158] Joel J. Wallman. 2018. Randomized benchmarking with gate-dependent noise. *Quantum* 2 (2018), 47.

[159] Joel J. Wallman and Joseph Emerson. 2016. Noise tailoring for scalable quantum computation via randomized compiling. *Phys. Rev. A* 94, 5 (2016), 052325.
[160] Dave Wecker, Matthew B. Hastings, and Matthias Troyer. 2015. Progress towards practical quantum variational algorithms. *Phys. Rev. A* 92, 4 (2015), 042303.
[161] Nathan Wiebe, Christopher Granade, and David G. Cory. 2015. Quantum bootstrapping via compressed quantum Hamiltonian learning. *New J. Phys.* 17, 2 (2015), 022005.
[162] Nathan Wiebe, Christopher Granade, Christopher Ferrie, and David Cory. 2014. Quantum Hamiltonian learning using imperfect quantum resources. *Phys. Rev. A* 89, 4 (2014), 042314.
[163] Nathan Wiebe, Christopher Granade, Christopher Ferrie, and David G. Cory. 2014. Hamiltonian learning and certification using quantum resources. *Phys. Rev. Lett.* 112, 19 (2014), 190501.
[164] Christopher J. Wood and Jay M. Gambetta. 2018. Quantification and characterization of leakage errors. *Phys. Rev. A* 97, 3 (2018), 032306.
[165] K. Wright, K. M. Beck, S. Debnath, J. M. Amini, Y. Nam, N. Grzesiak, J.-S. Chen, N. C. Pisenti, M. Chmielewski, C. Collins, et al. 2019. Benchmarking an 11-qubit quantum computer. arXiv:1903.08181. Retrieved from https://arxiv.org/abs/1903.08181.
[166] Jun Yoneda, Kenta Takeda, Tomohiro Otsuka, Takashi Nakajima, Matthieu R. Delbecq, Giles Allison, Takumu Honda, Tetsuo Kodera, Shunri Oda, Yusuke Hoshi, et al. 2018. A quantum-dot spin qubit with coherence limited by charge noise and fidelity higher than 99.9%. *Nat. Nanotechnol.* 13, 2 (2018), 102.
[167] Alwin Zulehner, Alexandru Paler, and Robert Wille. 2018. An efficient methodology for mapping quantum circuits to the IBM QX architectures. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 38, 7 (2018), 1226–1236.

Received March 2020; revised April 2021; accepted April 2021