


Early Fault-Tolerant Quantum Computing

Amara Katarawa,¹ Katerina Gratsea,^{1,2} Athena Caesura,¹ and Peter D. Johnson^{1,*}

¹*Zapata AI Inc., Boston, Massachusetts 02110, USA*

²*Institut de Ciències Fotòniques (IFCO), The Barcelona Institute of Science and Technology, Av. Carl Friedrich Gauss 3, Castelldefels, Barcelona 08860, Spain*

 (Received 12 January 2024; revised 31 March 2024; published 17 June 2024)

In recent years, research in quantum computing has largely focused on two approaches: near-term intermediate-scale quantum (NISQ) computing and future fault-tolerant quantum computing (FTQC). A growing body of research into early fault-tolerant quantum computing (EFTQC) is exploring how to utilize quantum computers during the transition between these two eras. However, without agreed-upon characterizations of this transition, it is unclear how best to utilize EFTQC architectures. We argue for the perspective that this transition period will be characterized by a law of diminishing returns in quantum error correction (QEC), where the ability of the architecture to maintain quality operations at scale determines the point of diminishing returns. Two challenges emerge from this picture: how to model this phenomenon of diminishing return of QEC as the performance of devices is continually improving and how to design algorithms to make the most use of these devices. To address these challenges, we present models for the performance of EFTQC architectures, capturing the diminishing returns of QEC. We then use these models to elucidate the regimes in which algorithms suited to such architectures are advantageous. As a concrete example, we show that for the canonical task of phase estimation, in a regime of moderate scalability and using just over one million physical qubits, the “reach” of the quantum computer can be extended (compared to the standard approach) from 90-qubit instances to over 130-qubit instances using a simple early fault-tolerant quantum algorithm, which reduces the number of operations per circuit by a factor of 100 and increases the number of circuit repetitions by a factor of 10 000. This clarifies the role that such algorithms might play in the era of limited-scalability quantum computing.

DOI: [10.1103/PRXQuantum.5.020101](https://doi.org/10.1103/PRXQuantum.5.020101)

I. INTRODUCTION

Quantum computers were first proposed to efficiently simulate quantum systems [1]. It then it took about a decade before it was discovered that quantum phenomena, such as superposition and entanglement, could be leveraged to provide an exponential advantage in performing tasks unrelated to quantum mechanics [2]. Although of no practical use, the Deutsch-Jozsa algorithm sparked interest in using a quantum computer to perform other tasks beyond simulating quantum systems [3,4], the most famous case being Shor’s algorithm [5]. Around the same time, the ground-breaking discovery of quantum error-correcting codes (QECCs) [6–10] set the stage for practical quantum computing. This has shown that errors due to

faulty hardware could not only be identified but also corrected. Two pieces of the puzzle have been left, namely:

- (1) Could quantum computation be done in a fault-tolerant manner, i.e., could error-corrected qubits perform better than physical qubits?
- (2) Can one rigorously prove the existence of a threshold [11] below which error can be reduced exponentially in the time and memory overhead cost?

The first piece of the puzzle was tackled by Peter Shor [12] and later, building on his work, threshold theorems have been proved assuming various kinds of error models [13–15]. For a specific quantum error-correcting code and a noise model, it is then left to prove and find error thresholds, with early works being Refs. [16–19]; this continues to be an active area of research [20–23].

Meanwhile on the hardware side, astonishing progress has been made across various modalities (e.g., superconducting, ion-trap, photonic, etc.) in terms of extending qubit coherence times and improving entangling operations [24–31]. Driven by such advances, a watershed moment occurred in 2016 when IBM put the first quantum

*Corresponding author: peter.d.johnson22@gmail.com

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article’s title, journal citation, and DOI.

computer on the cloud, giving the public access to quantum computers. This event spurred widespread interest in finding near-term quantum algorithms that did not need the full machinery of fault tolerance. These algorithms first formulate the problem as a solution to the ground state of some Hamiltonian, store a trial ansatz on the quantum processing unit (QPU), and use a classical optimizer to find the optimal parameters. The variational principle guarantees that the optimized parameters will produce a state the energy of which upper bounds that of the target Hamiltonian. These so-called hybrid *classical-quantum* algorithms allow one to use short-depth quantum circuits and reduce the need for high-quality quantum coherence. They have found application in areas of quantum chemistry [32], machine learning [33,34], and optimization [35]. Another watershed moment occurred when the Google Quantum AI team, along with collaborators, announced their achievement of so-called “quantum supremacy” [36]; they argued that their hardware accomplished a computational sampling task far faster than possible with available supercomputers.

Despite this progress, there is still need to reduce errors and the area of quantum error mitigation arose as attempts have been made to meet the needs of these applications [37–41]. This way of using a QPU is what is characteristic of the so called *NISQ era* [42]. Although there is no strict definition of what constitutes a NISQ device, it can generally be assumed that NISQ devices are too large to be simulated classically but also too small to implement quantum error correction. IBM’s work [43] is in some sense the true dawn of the NISQ era, i.e., a quantum device where error mitigation is important and classical simulation is hard. A flurry of work [44–46] immediately arose pushing classical methods of simulation and claiming to reproduce IBM’s results. This is a new phase in which NISQ devices will be put to the test by state-of-the-art classical simulators and vice versa. Eventually, as the system size under study is increased, the corresponding Hilbert space for the quantum systems grows exponentially and the NISQ device will be the only viable simulation approach. There are methods in this regime that can extend the size of simulations by cutting up the circuit and running smaller versions on smaller devices and then knitting the results back with some classical postprocessing [47].

But an important question remains and, in a very obvious sense, the elephant in the room is, “Are NISQ devices and NISQ algorithms up for the task of realizing quantum advantage at utility scale?” Work has been done in quantum chemistry where the problem can be precisely asked in, e.g., finding the ground state of large molecules. The best estimates so far for resource estimates suggest that the variational quantum eigensolver (VQE) is not up to the task [48]. Other work suggests a possible quantum advantage for the quantum adiabatic optimization ansatz (QAOA) [49–51] in optimization but it remains to be seen

whether these claims can be confirmed in the presence of noise at scale.

Given these roadblocks, should our attitude be to wait for fully fault-tolerant devices? An area of research offers an intriguing possibility; we are offered a trade-off—we require fault-tolerant quantum computing but the ability to run smaller quantum circuits at the cost of requiring more sampling for the quantum device. Such a trade-off has been the focus of a substantial amount of research in the past few years [52–61]. However, in a regime where we are able to arbitrarily scale the number of physical qubits while maintaining quality fault-tolerant protocols, such a trade-off would not be favorable; by increasing the circuit size using methods such as quantum amplitude amplification [62], the additional overhead of efficient fault-tolerant protocols is negligible compared to the overall reduction in run time. Accordingly, such a trade-off would be better suited to a setting in which the efficiency of fault-tolerant protocols worsens with increasing system size. If the ability to scale the number of physical qubits (i.e., the “scalability”) is compromised by a worsening of the operations, then these diminishing returns will, in turn, limit the size of problems that can be solved. Such a regime of computation has been referred to as *early fault-tolerant quantum computing* (EFTQC) [64], a natural successor to the NISQ era. A field of research has emerged recently where the proposed quantum algorithms enable this “circuit-size versus sample-cost” trade-off [52,54,55,57,58,65–69]. Two questions are then placed before us:

- (1) Will this regime of limited-scale quantum computers exist in a meaningful way?
- (2) If so, will we be able to unlock intrinsic quantum value at scale in this regime?

The ultimate answers to these questions will depend on hard-to-predict factors, including hardware, quantum error-correcting codes, quantum algorithm advances, and improvements in competing classical hardware and algorithms. Rather than predicting the time line of these advances, we propose a quantitative framework to track their progress. However, rather than attempt to precisely model device characteristics, we will aim to model the performance of quantum error correction by abstracting the critical hardware features that predominantly affect the logical error rates of quantum error correction. In Fig. 1, we depict the landscape in which this framework assesses the ability of a given hardware vendor to supply useful physical qubits and operations, transitioning from NISQ to EFTQC to FTQC.

To address the first question, we propose a very simple model [see Eq. (1)] to quantitatively discuss these regimes. This simple model describes how the quality of elementary quantum operations degrades as the system size is increased; i.e., we model the physical gate error rate as a

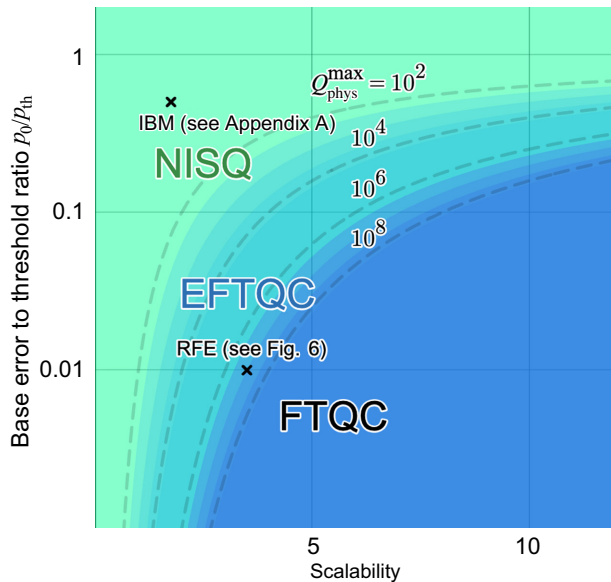


FIG. 1. A rough demarcation of the regimes of NISQ, EFTQC, and FTQC according to the scalability model introduced in Sec. II. The vertical axis quantifies the base error rate (i.e., that achievable for a single qubit), while the horizontal axis quantifies the ability of the architecture to maintain low error rates as it is scaled (i.e., its *scalability*). The contours indicate the maximum physical-qubit number that the architecture is warranted in scaling to as predicted by the scalability model of Eq. (1). The NISQ-to-EFTQC transition is characterized by having enough qubits to implement fault-tolerant non-Clifford operations (e.g., T factories), while the EFTQC-to-FTQC transition is characterized by the ability to accommodate very large problem instances (e.g., encoding 10 000 logical qubits using in 10^9 physical qubits). The red cross corresponds to data presented in Appendix A, which estimates that a hardware vendor of today (IBM) has a scalability of 1.75 with $p_0 = 0.005$. An editable version of the plot can be accessed at Ref. [63].

function of the physical-qubit number. We dub this model the *scalability* of a device. We emphasize that this model drastically simplifies the complex nature of device modeling. However, the models that we propose can be understood as slight generalizations of those typically used for fault-tolerant resource estimation [70,71]. For the second question, we quantify how recently developed algorithms can extend the “reach” of quantum computers with limited *scalability*. This is an important step toward understanding what value such methods can provide.

The paper is organized as follows. In Sec. II, we present the scalability model and apply this to an example resource estimation for the quantum phase-estimation (QPE) algorithm. In Sec. III, we review progress in algorithms for early fault-tolerant quantum computers and then present an example of one such algorithm, showing how it can improve the capabilities of a device with limited scalability. Finally, in Sec. IV, we discuss the implications of

our findings and outline important future research directions.

II. MODELING EARLY FAULT-TOLERANT QUANTUM COMPUTATIONS

A. Introduction to the scalability model

In this section, we establish and discuss the precise sense in which a device can be an *early fault-tolerant device*. We first note the tension in the very phrase *early fault tolerance*. Fault tolerance evokes the ability to ensure efficient suppression of error despite the use of faulty operations [12]. The string of results [9,13,14,18] collectively known as the threshold theorems show that *in principle* this can be achieved. In fact, due to these results, we know [13] that under quite general assumptions, such as allowing for long-range correlations of noise and non-Markovianity, fault tolerance is still possible. These foundational works would put the threshold error rate around 10^{-5} – 10^{-6} . However, more optimistic threshold predictions have been made using numerical investigations [72]. For the surface code [73], which is a leading contender for practical quantum computing [74], such simulations have led to the prediction of quite optimistic thresholds of approximately 1% [75], which have also been argued for analytically [76]. On the other hand, numerical thresholds are based on particular assumptions of noise and error that cannot fully capture the complexity of quantum architectures at scale. For example, an important assumption is that a single number can be used to capture the performance of operations and that this single number remains constant as larger code distances [77] are used [75]. Such thresholds have become the established targets for hardware developers [78–81].

The “early” in “early fault tolerance,” on the other hand, suggests some kind of limited ability to achieve fault tolerance, i.e., using a polynomial amount of resources to achieve exponential error suppression [82,83]. This tension is what lies behind the motivation for this work.

A key insight toward resolving this tension is to realize what we call the *scalability requirement*:

In order to reap the benefits of being below any threshold, an approach to building a quantum architecture must be able to maintain each operation below the threshold error rate as larger and larger architectures are built.

The failure to achieve the scalability requirement implies the existence of scale-dependent errors. While NISQ devices fail to meet the scalability requirement, the hope is that future quantum computers will meet this requirement over a large range of device scales, effectively enabling fault-tolerant quantum computing. Taking EFTQC to be the transition from NISQ to FTQC, we will characterize it as satisfying the scalability requirement only over a limited range of device scales. Ultimately, this investigation is motivated by wanting to understand the

prospects of using early fault-tolerant quantum computers to solve utility-scale problems. In order to do so, we must make use of some model of early fault-tolerant quantum computers. While accurate modeling of such quantum computers might include modality-specific considerations such as the heterogeneity of superconducting qubit chiplet architectures [84] or connecting elementary logic units of ion-trap architectures [85], we will take a modality-independent approach. The reason for this choice is to enable the setting of requirements for solving utility-scale problems that any hardware vendor can aim to satisfy. To achieve this, our model will describe the simplest characteristics necessary to determine the effectiveness of quantum error correction, as the quality of the logical operations determines the algorithmic capabilities of the architecture.

While we have cast our modeling as a simplified approach, there is an important sense in which it generalizes previous approaches to modeling fault-tolerant quantum computations. These approaches to predicting the physical-qubit costs and run times for a quantum computation assume that device error is scale independent [70,71,86]. Accordingly, these approaches do not model the transition between NISQ devices with scale-dependent error and future FTQC with nearly scale-independent error. So our challenge is expand these models to accommodate quantum computations in the regime in which the scalability requirement is only satisfied up to a point or, in other words, to model the performance of quantum computations in devices that have scale-dependent error rates.

Therefore, our approach will be seen as (1) a generalization that incorporates both the scale-independent and -dependent settings and (2) an attempt to bridge the observed scale dependence of error in today’s devices with the hoped-for scale independence of error in future quantum architectures. We expect that the degree of scale dependence will inform the capabilities of the architecture being modeled. Furthermore, scale-dependent error may warrant the development and use of quantum algorithms that are suited to this limitation. These considerations motivate the main question pursued in the remainder of this paper: “How does the degree of scale-dependent error determine the capabilities of early fault-tolerant quantum computers?”

To address the origin of scale-dependent error, we consider the general setup used to prove the fault-tolerant threshold theorems. It is assumed that we have the following Hamiltonian as

$$\mathcal{H} = \mathcal{H}_S + \mathcal{H}_B + \mathcal{H}_{SB},$$

where \mathcal{H}_S is the Hamiltonian governing the evolution of the system, which for our discussion can be the evolution corresponding to implementing the quantum gate, \mathcal{H}_B governs the evolution of some bath, and \mathcal{H}_{SB} entangles the bath with the qubits in the computation. The

scale-dependent errors arise from the engineering details involved in implementing \mathcal{H}_S as larger and larger chips are developed. These engineering problems cannot be completely inserted into \mathcal{H}_{SB} and yet would ultimately impact how easily we could stay below threshold as we try to scale up. For fixed frequency qubits in superconducting architectures, the issue of “frequency crowding” affects the quality that any single two-qubit gate can achieve [87]. The number of frequencies that must be avoided when implementing the cross-resonant gate increases as the number of qubits increases in the chip; this makes targeting the required frequency harder and harder as you scale up. Another scale-dependent engineering difficulty can arise from unwanted interactions between control lines going into the chip. The calibration of these pulses is partly a classical problem that gets more complicated and cumbersome as the chip gets larger. The issue of “cross-mode coupling” at ion traps will affect the fidelity of the gate [88,89], where the target has a specific motional mode but unwanted couplings destroy the quality of the gate. The problem has a classical component that scales with the number of qubits. In the above cases, the physics of accurate addressability of a qubit or pairs of qubits is a problem that becomes harder with an increasing number of qubits and thus affects the quality of the gate operation. Recent works have explored the consequences of scale-dependent errors, which would most likely arise from the limited resources to control qubits and design good-quality operations [90,91].

It is reasonable to believe that the assumption of scale-independent error rates may eventually become effectively true on account of modularity, as future quantum architectures will likely be made from repeated modular components [93,94]. And while the holy grail of (effectively) scale-independent subthreshold error rates may some day be realized, quantum architectures will necessarily undergo a transition from today’s scale-dependent error to the future of scale-independent error. We will take this transition to be the defining characteristic of early fault-tolerant quantum computing.

We start by describing the particular setting in which we model scale-dependent error. Our model will center around the concept of *scalability*, the ability to maintain low error rates (e.g., subthreshold) as larger architectures are requested. Our setting and model are driven by the need to answer the following question: “For a *series* of quantum computations of increasing size, how well will a hardware vendor be able to service the request to run the quantum computations?” Accordingly, we will not consider the capabilities of a single quantum device or a single quantum architecture, as the hardware vendor might have several architectures to service computations of various sizes. Furthermore, we will not consider the capabilities of the hardware vendor as they improve over time, as our hypothetical test is used to assess capability at one moment

in time. In order to make this quantitative, we can consider a *scalability profile*: an empirically derived function that reports the worst-case error rate among the elementary operations of the device as a function of the requested number of physical qubits. For the case of today’s IBM devices, we present data on their scalability profile in Appendix A. In lieu of scalability-profile data for future quantum vendors, we propose a simple parametrized model for this function:

$$p_{\text{phys}}(Q_{\text{phys}}; \mathcal{V}) = p_0 Q_{\text{phys}}^{1/s}, \quad (1)$$

where Q_{phys} is the number of physical qubits in the architecture and \mathcal{V} labels the particular hardware vendor that is providing the qubits at any time. Parameters p_0 and s capture the base error rate and the “scalability,” respectively. It is helpful to view this model as a power-law fit of a scalability profile. In Appendix B, we investigate the more optimistic case of a logarithmic model. The case of $s = \infty$ corresponds to scale-independent error, or infinite scalability, while any finite value of s corresponds to the case of finite or limited scalability.

We would like to emphasize here that the scalability model proposed is a first attempt to approach the problem of the scalability profile of future quantum hardware. This is a very hard and complex problem and it is highly hardware dependent. However, the goal of this work is to show that given some tentative assumptions on the hardware scalability profile, resource-estimation tools could help to explore and evaluate the consequences on quantum computing research and development.

As we will show in Sec. III, in the context of fault-tolerant quantum computing, a finite scalability will result in a finite limit on the number of physical qubits being used before fault-tolerant protocols yield diminishing returns. We then explain how this limit on the physical-qubit number places a limit on the problem sizes that the architecture can accommodate. Importantly, all of these considerations apply in the setting where fault-tolerant protocols are being used. This differs from the setting assumed for NISQ quantum computing [42], where physical qubits instead of logical qubits are used for computation. Before moving to Sec. III, we provide some perspective on the transition from the NISQ regime to the EFTQC regime. Specifically, in the rest of this subsection, we estimate the minimal number for Q_{phys} in an EFTQC computation, assuming a simple surface-code architecture.

The total number of physical qubits for a computation can be written as $Q_{\text{phys}} = Q_{\text{comp}} + Q_{\text{MSD}}$, where Q_{comp} is the number of physical qubits used to compute (i.e., to store and route the logical data) and Q_{MSD} are the physical qubits used for magic state distillation. To calculate the minimum number of qubits required for QEC, we will set $Q_{\text{comp}} = 2(d+1)^2$ [71], corresponding to a single surface-code logical qubit, and pick the smallest

magic state factories that give an improvement on the error rate.

The most efficient magic state factories known in the surface code are given in Ref. [95]. We have listed the smallest of these in Table I (note that these do not give much of an improvement over the physical error rate). Ignoring the errors introduced from the gates involved in the injection process, the error rate of an injected T state is given by $\max(p_{\text{out}}, p_L)$. Thus, in both of the cases shown in Table I, we expect that the error of the injected T state will be p_L . This means that in an EFTQC demonstration with minimal distance ($d = 3$), the error rate for the computation with high-quality operations will be 10^{-5} , a factor-of-10 decrease in the failure rate. In contrast, operations of lower quality do not exhibit any improvement over the physical error rate, as the distance 3 is trivially small for operations of this quality. If we have a single logical qubit, then this minimum viable example of EFTQC will be at least 554 and 842 qubits depending on the quality of the physical operations.

Note that the magic state distillation factory has dominated the number of qubits. As a result, the FTQC community has put a lot of work into decreasing the size of factories [95], improving injection protocols [96], or eliminating distillation entirely [97]. One would expect that the first EFTQC demonstrations will employ many of these techniques rather than the “pure-FTQC” calculation presented above. In a more careful calculation to estimate a lower bound for the EFTQC range, one may want to take such techniques into account and calculate Q_{phys} . Refining this estimate to clarify and lower the NISQ-to-EFTQC transition is important future work.

B. Example: Quantum phase estimation compiled to the surface code

In Sec. II A, we introduced Eq. (1) as a model for how physical operation error rates might increase with the system size. To understand the implications of this model, we work through the example of using the QPE algorithm [62] to solve the phase-estimation task. The task of phase estimation is to estimate the eigenphase of a unitary operator U with respect to an eigenstate $|\psi\rangle$, assuming access to circuits that implement $c-U$ and prepare $|\psi\rangle$. We review how to estimate the quantum resources required to perform this task under the scalability model and compare these to the ideal model case (i.e., $s \rightarrow \infty$).

A fault-tolerant resource estimation answers the following question: “How many physical qubits are needed per logical qubit to ensure that the logical error rates are sufficiently low to make the algorithm succeed (with some probability)?” To answer this, we must (1) determine what logical error rates the algorithm deems as “sufficiently low” and (2) establish the relationship between the logical error rate and the quantum resources.

For (1), the QPE algorithm will succeed with sufficiently high probability as long as the total circuit error rate is below some value p_C . We will set $p_C = 0.1$, noting that, in the literature, this tolerable circuit error rate varies from 0.1 [71] to 0.01 [70] but can be made lower using alternative algorithms [99–101]. This tolerable circuit error rate, along with the number of operations per circuit, lets us bound the tolerable operation error rate. The quantum circuit will ultimately be compiled into a set of logical operations that are implemented using fault-tolerant protocols [e.g., initialization of $|0\rangle$, measurement in the computational basis, H gate, controlled-NOT (CNOT) gate, and T gate]. We define G_C to be the number of elementary logical operations (including idling [102]) used by the circuit. To ensure that the circuit error rate is less than p_C , it suffices [103] to ensure a logical error rate of $p_L \leq p_C/G_C$ (by the union bound).

For the quantum phase-estimation algorithm, G_C is determined by the target accuracy and the number of operations per c - U . To yield an estimate of the phase angle to within ϵ of the true value requires the use of a circuit with $1/\epsilon$ applications of c - U [104]. For our purposes, we assume a model for G_C by fitting data in Table II of Ref. [64] to the following power law, where that work has chosen ϵ to be approximately half a percent of the total system energy,

$$G_C = \alpha Q_L^\beta, \tag{2}$$

yielding $\alpha = 4.12 \times 10^9$ and $\beta \approx 0.515$. Thus, the success of the algorithm is ensured (with high probability) by

$$p_L \leq \frac{p_C}{\alpha Q_L^\beta}. \tag{3}$$

For simplicity, we shall assume that the number of logical qubits needed for magic state factories is accounted for in this model (for details of the assumptions and the relevant references, see the notes in the Ref. [98] desmos plot of Fig. 3) and we will assume that the physical-qubit overhead is captured by the code distance used for the data qubits (though the factories typically have multiple layers of concatenation with differing code distance).

For (2), we will assume a model of error suppression based on simulations of the surface code in Ref. [105]. This

TABLE I. The two smallest possible magic state distillation factories given by Ref. [95]: p_{phys} is the physical error rate, Q_{phys} is the number of physical qubits required to create the factory, p_{out} is the probability that the output state magic state is incorrect, $Q_{\text{min, EFTQC}}$ is a rough lower bound on the number of qubits in an EFTQC calculation, and p_L is the logical failure rate in that lower bound calculation, where $d = 3$.

Quality of operations	Factory name	p_{phys}	Q_{phys}	p_{out}	$Q_{\text{min, EFTQC}}$	p_L
Higher-quality	(15-to-1) _{5,3,3}	10^{-4}	522	4.7×10^{-6}	554	10^{-5}
Lower-quality	(15-to-1) _{7,3,3}	10^{-3}	810	5.4×10^{-4}	842	10^{-3}

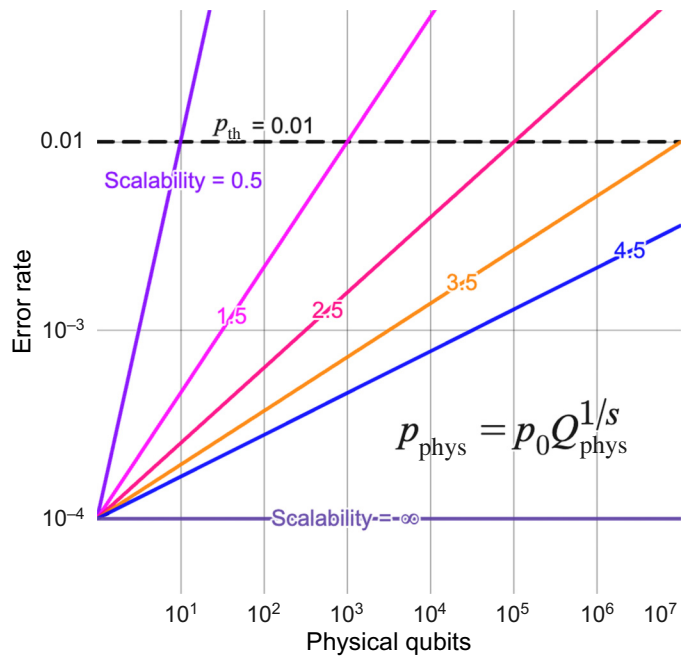


FIG. 2. The concept of scalability captures the ability of a quantum architecture to maintain low physical error rates as the number of physical qubits of the architecture is increased. Here, we show the scalability profiles of different quantum architectures given by the scalability model [Eq. (1)] for different scalability values ($s = 0.5, 1.5, 2.5, 3.5, 4.5, \infty$) and base error rate $p_0 = 10^{-4}$. A finite scalability implies that beyond a certain physical-qubit size, the architecture cannot maintain physical error rates below the error threshold (p_{th}) of the fault-tolerant protocol. An editable version of the plot can be accessed at Ref. [92].

model is

$$p_L = A(p_{\text{phys}}/p_{\text{th}})^{(d+1)/2}, \tag{4}$$

where Ref. [71,105] estimates $A = 0.1$ and $p_{\text{th}} = 0.01$. The number of physical qubits used to encode one logical qubit in the surface code is $2(d + 1)^2$, leading to

$$Q_{\text{phys}} = 2(d + 1)^2 Q_L. \tag{5}$$

In the case that p_{phys} is independent of the number of physical qubits, p_L can be made arbitrarily small, with cost (depending on code distance d) scaling as $d \sim \log(1/p_L)$.

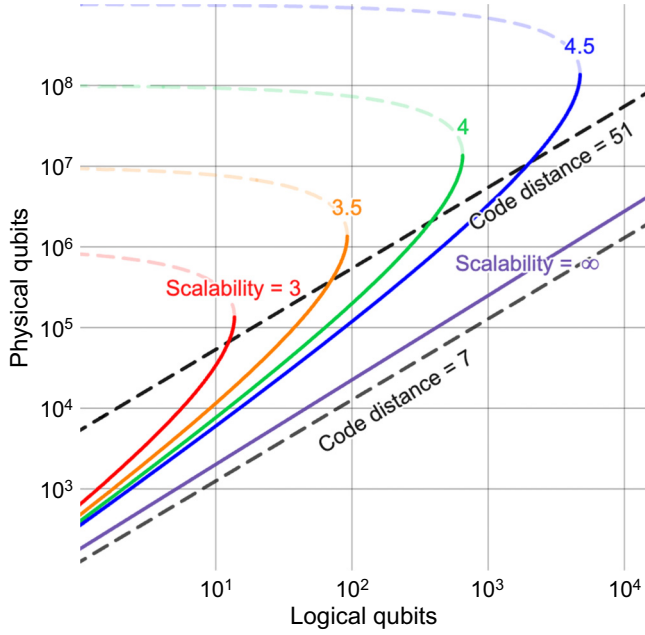


FIG. 3. The scalability model of Eq. (1) predicts that for each finite value of scalability parameter s , there is a maximum problem-instance size that can be accommodated by the architecture. Each curve is a contour in the $Q_{\text{phys}}-Q_L$ plane of a solution to Eq. (7) for a particular value of the scalability parameter s (3, 3.5, 4, 4.5, ∞). The remaining parameters of Eq. (7) are set to $p_{\text{th}} = 10^{-2}$, $p_0 = 10^{-4}$, $\alpha = 4.12 \times 10^9$, and $\beta = 0.515$ [see Eq. (2)], following Ref. [64, Table II]. The transition from solid to faded dashed curves occurs when the physical-qubit number reaches $Q_{\text{phys}}^{\text{opt}} = Q_{\text{phys}}^{\text{max}}/e^2$, beyond which increasing the code distance leads to diminishing returns. The diagonal black dotted lines show the physical-qubit count for two fixed code distances: 7 (small distance) and 51 (large distance). Note that code distance is discrete, which, if taken into account, would result in the contours jumping from one fixed-code-distance line to the next. However, we have chosen to allow for the distance parameter to be continuous, for ease of viewing the trends of the contours. An editable version of the plot can be accessed at Ref. [98].

However, if we replace p_{phys} with the Q_{phys} -dependent function $p_{\text{phys}}(Q_{\text{phys}})$ of Eq. (1) (i.e., the scalability model), the logical error rates cannot be made arbitrarily small. The smallest error rate is achieved when $p_{\text{phys}} = p_{\text{th}}$, which occurs when $Q_{\text{phys}} = (p_{\text{th}}/p_0)^s$; including more qubits (i.e., increasing the code distance) will lead to a decrease in the logical error rate. This number of physical qubits is therefore the maximal number of physical qubits that should be used under the scalability model:

$$Q_{\text{phys}}^{\text{max}} = (p_{\text{th}}/p_0)^s. \quad (6)$$

So, for example, when $p_{\text{th}} = 0.01$ and $p_0 = 0.001$ (as is sometimes assumed for superconducting qubit resource estimates with the surface code [71]), we have $Q_{\text{phys}}^{\text{max}} = 10^s$. A more optimistic setting of $p_0 = 0.0001$ leads to $Q_{\text{phys}}^{\text{max}} =$

10^{2s} . In Fig. 1, we depict contours of $Q_{\text{phys}}^{\text{max}}$ in the plane of p_0/p_{th} versus s .

The above concepts can be summarized as follows:

Requirement:

$$p_C \geq G_{CPL} \quad (\text{algorithm error tolerance})$$

Cost:

$$Q_{\text{phys}} = 2(d+1)^2 Q_L \quad (\text{surface-code overhead})$$

Models:

$$G_C = \alpha Q_L^\beta \quad (\text{QPE-circuit gate count})$$

$$p_L = A \left(\frac{p_{\text{phys}}}{p_{\text{th}}} \right)^{\frac{d+1}{2}} \quad (\text{surface-code logical error rate})$$

$$p_{\text{phys}} = p_0 Q_{\text{phys}}^{1/s} \quad (\text{scalability of physical error-rate model})$$

Putting these together, we can determine the number of physical qubits required to ensure that QPE returns an ϵ -accurate estimate (with high probability) as a function of the number of logical qubits Q_L (roughly corresponding to the problem size). This relationship is expressed by $Q_{\text{phys}}-Q_L$ pairs that ensure that Eq. (3) is satisfied (i.e., that the logical error rates are low enough for the algorithm to succeed):

$$\sqrt{8Q_L} \log \left(\frac{A\alpha}{p_C} Q_L^\beta \right) \leq \sqrt{Q_{\text{phys}}} \log \left(\frac{p_{\text{th}}}{p_0} Q_{\text{phys}}^{-1/s} \right). \quad (7)$$

Before applying this result to the quantitative example that has been set up, we make a few general remarks that apply to any algorithm analyzed in this manner.

First, we consider the right-hand side of this inequality. This function will determine an optimal value for Q_{phys} , which we label as $Q_{\text{phys}}^{\text{opt}}$. Previously, we have described a maximum value of Q_{phys} as set by the condition of p_{phys} being below threshold. However, the maximum allowed value of Q_L is now set by a function of Q_{phys} ; to increase this ceiling, we should maximize the right-hand-side function of Q_{phys} . This function achieves its maximum of $(2e/s)^2 (p_{\text{th}}/p_0)^s$ at a value of

$$Q_{\text{phys}}^{\text{opt}} = \frac{1}{e^2} \left(\frac{p_{\text{th}}}{p_0} \right)^s \leq Q_{\text{phys}}^{\text{max}}. \quad (8)$$

This is considered the optimal number of physical qubits in that it enables the use of the largest number of logical qubits. As an example, for $p_{\text{th}} = 0.01$, $p_0 = 0.0001$, and $s = 3.5$, the optimal number of physical qubits is $Q_{\text{phys}}^{\text{opt}} \approx 1.35 \times 10^6$.

These quantities of $Q_{\text{phys}}^{\text{max}}$ and $Q_{\text{phys}}^{\text{opt}}$ can help us to quantify the scalability parameters p_0 and s that are relevant to the NISQ-to-EFTQC transition and the EFTQC-to-FTQC transitions. At the end of Sec. II A, we have described how the NISQ-to-EFTQC transition might occur in the range of 100–10 000 physical qubits. Considering Eqs. (6) and

(8), this determines the (p_0, s) pairs characteristic of this transition and shown as the red-to-green blend in Fig. 1.

We motivate the idea that the transition from EFTQC to FTQC is characterized by how the quantum computations are “bottlenecked.” In the case of fault-tolerant quantum computing, it is envisioned that the ability to run larger and larger quantum computations is possible as long as the computations are not practically limited by resources such as time and energy. We propose that early fault-tolerant quantum computing be characterized by the regime in which the largest possible quantum computations are limited by the maximum number of physical qubits warranted in the architecture ($Q_{\text{phys}}^{\text{max}}$ or $Q_{\text{phys}}^{\text{opt}}$). Viewing time as the limiting resource, if we assume that the quantum computation must finish within a month, then this limits the problem sizes that can be accommodated accordingly. Using the quantum chemistry resource estimations of Ref. [71] as a point of reference, problem instances that would take a month would require on the order of 10^7 physical qubits. There may be other classes of problems that become run-time limited when fewer or more physical qubits are required. Thus, in Fig. 1, we depict the transition from EFTQC to FTQC as the green-to-blue gradient ranging from 10^6 to 10^8 .

Second, we consider the left-hand side of Eq. (7). Most of the parameters are contained in the factor $A\alpha/\epsilon p_C$. In Sec. III B, we will explain the importance of this factor in quantifying the “burden” placed on the elementary fault-tolerant protocols. Equation (7) shows that decreasing this burden factor affords a decrease in the number of physical qubits Q_{phys} . Alternatively, when fixing the number of physical qubits, a reduction in the burden factor affords an *increase* in the number of logical qubits and subsequently the maximum problem size or “reach” of the quantum computer. The methods introduced in Sec. III A will be understood to reduce this burden factor, enabling algorithms to be run using fewer physical qubits, though at the cost of an increase in run time.

In Fig. 3, we show the contours of solutions to Eq. (7) for several scalability values s . The most striking feature is that for the finite values of scalability ($s < \infty$), there is a maximum-size instance (measured by Q_L) that the architecture can accommodate using the QPE algorithm. For example, in the case of $s = 3.5$, $p_0 = 0.0001$, and $p_{\text{th}} = 0.01$, we find that the largest instance that can be accommodated (i.e., the “reach” of the quantum architecture) is $Q_L \approx 90$. The maximum number of logical qubits Q_L^{max} can be solved for by setting $Q_{\text{phys}} = Q_{\text{phys}}^{\text{opt}}$ in Eq. (7) and solving for Q_L ,

$$Q_L^{\text{max}} = \frac{Q_{\text{phys}}^{\text{opt}}}{8s^2\beta^2 W\left(\sqrt{\left(\frac{A\alpha}{p_C}\right)^{\frac{1}{\beta}} \frac{Q_{\text{phys}}^{\text{opt}}}{8s^2\beta^2}}\right)^2}, \quad (9)$$

where $W(x)$ is the solution to $W(x)\exp(W(x)) = x$, known as the Lambert W function. Using the upper bound of $W(x) \leq \ln(x)$, we can lower bound the maximum qubit number as

$$Q_L^{\text{max}} \geq \frac{\left(\frac{p_{\text{th}}}{p_0}\right)^s}{2e^2s^2\beta^2 \ln\left(\left(\frac{A\alpha}{p_C}\right)^{\frac{1}{\beta}} \frac{\left(\frac{p_{\text{th}}}{p_0}\right)^s}{8e^2s^2\beta^2}\right)^2}, \quad (10)$$

where we have used the expression for $Q_{\text{phys}}^{\text{opt}}$. This maximum solvable problem size motivates the question explored in the next section: “With a fixed scalability, is it possible to extend the ‘reach’ of a quantum architecture using algorithms designed for finite scalability?”

III. QUANTUM ALGORITHMS FOR EARLY FAULT-TOLERANT QUANTUM COMPUTERS

A. Review of methods

Section II ended with a question about how to extend the reach of quantum computers that have limited scalability or, in other words, early fault-tolerant quantum computers. A growing body of work in quantum algorithms has developed a suite of methods that might be used to address this problem. Such algorithms solve the tasks of phase estimation [100,106–108], multiple-eigenvalue estimation [101,109–111], ground-state property estimation [112], amplitude estimation [52,54,55,116], ground-state energy estimation [57,60,61,66,67,69], and ground-state preparation [59,115]. In this subsection, we provide a nonexhaustive review of the literature in this area. Note that some of the works do not use the term “EFTQC.” Nevertheless, we include them because of their influence on later works [52,54,55] or the similarity in their motivations [61,111].

These algorithms have typically been developed with certain improvements in mind that include: reduction of the logical qubit number [66,115], reduction of the number of operations per circuit [52,54,57], the reduction of expensive operations [64,67] (e.g., non-Clifford operations such as T gates and Toffoli gates), and establishing or increasing the robustness to error [54,100,101,107,117]. In many cases, achieving these improvements comes at a cost. The predominant cost is an increase in the number of circuit repetitions (also known as the “sample complexity,” “number of samples,” or “shots”), and, subsequently, the run time. Another cost is an increase in classical processing (e.g., converting the measurement-outcome data from the many circuit repetitions into the estimate of the ground-state energy). In Sec. III B, we will detail an example algorithm where these trade-offs can be easily understood.

One of the first algorithms suited for early fault-tolerant quantum computers was the so-called α -VQE method [52].

This method for solving the task of amplitude estimation enables a trade-off between the number of quantum operations per circuit $O(1/\epsilon^\alpha)$ and the number of circuit repetitions $\tilde{O}(1/\epsilon^{2(1-\alpha)})$ (where \tilde{O} indicates that we ignore polylog factors), set by a tunable parameter α . Later, Wang *et al.* introduced a variable-depth amplitude estimation algorithm that is robust to substantial amounts of circuit error [54]. Similar methods have been explored in the context of quantum algorithms for finance [55,118] and some have been implemented on quantum hardware [119,120].

Another thread in the development of quantum algorithms for early fault-tolerant quantum computing has focused on problems related to physical systems such as electronic structure or condensed-matter systems specified by their Hamiltonians. In this direction, one of the first papers to introduce the phrase “early fault-tolerant” was Ref. [64], where the author reduced the counts of expensive non-Clifford operations to make simulation of the Fermi-Hubbard model more amenable to smaller fault-tolerant quantum computers. Other methods have sought to reduce the logical qubit requirements. Previous approaches had mostly been based on the QPE algorithm [15], which uses additional ancilla qubits for reading out the phase [121]. A method for estimating the spectrum of a Hamiltonian without the use of QPE (and its ancilla qubit overhead) was introduced in Ref. [122]; instead, the spectrum is estimated by classically postprocessing measurement-outcome data from Hadamard tests of the $c\text{-}e^{iHt}$ circuit. Then, one of the first papers to motivate their algorithm development in the context of early fault-tolerant quantum computers was Ref. [66]. The authors developed a novel postprocessing technique for the measurement-outcome data generated in Ref. [122] and carried out an analysis of their ground-state energy-estimation (GSEE) algorithm, showing that they could achieve a run time with Heisenberg-limit scaling of $O(1/\epsilon)$, compared to the $O(1/\epsilon^4)$ run time of Ref. [122] (when improved and applied to the task of GSEE). Placing their work in the context of early fault-tolerant quantum computing, this work solidified this new research direction in quantum algorithms and helped place earlier works in the context of early fault-tolerant quantum computing. For a summary of the performance of this and related ground-state energy-estimation algorithms, see Table III. By combining the insights of Ref. [66], linear combination of unitaries [123], and QDRIFT [124], in Ref. [67], the authors have developed a method to exploit structure in the Hamiltonian to make the overall complexity of GSEE independent of the number of terms in the Hamiltonian. They have also demonstrated that their method enables trading the number of operations per circuit for number of circuit repetitions. A methodology similar to the above works has been applied to the task of estimating ground-state properties [112], which is often required in industrially relevant quantum chemistry calculations [106,125].

TABLE II. Quantum algorithms for EFTQC.

Task	Algorithm
Phase estimation	Time-series estimator [106]
	Randomized Fourier estimation [100,107]
	Modified robust phase estimation [108]
Multiple-eigenvalue estimation	Adaptive multiorder phase estimation [109]
	Robust multiple-phase estimation [101]
	Multimodal multilevel QCELS [110]
	Observable dynamic mode decomposition [111]
Ground-state property estimation	Heaviside-filter property estimation [112]
Amplitude estimation	α -QPE [52]
	Robust amplitude estimation [54]
	Power-law AE (also QoPrime AE) [55]
Ground-state energy estimation	Quantum filter diagonalization [53]
	Multireference selected quantum Krylov [113]
	Variational QPE [56]
	Fourier filtering [66]
	Statistical phase estimation [67]
	Gaussian filter [57]
	QCELS [60]
	Quantum Lanczos [61]
	Rejection sampling [69]
	Gaussian QPE [114]
	Gaussian booster [59]
Ground-state preparation	Quantum eigenvalue transform of unitary matrices with real polynomials [58]
	Single-ancilla Lindbladian [115]

The EFTQC algorithms developed for amplitude estimation [52,54,55] have established a trade-off between operations per circuit and circuit repetitions. These result in tuning the run time between Heisenberg-limit scaling $O(1/\epsilon)$ and the central-limit scaling $O(1/\epsilon^2)$. This raises the question of whether a similar trade-off can be established for the task of ground-state energy estimation. Such circuit trading has been established in Ref. [57]. The authors have shown an exponential reduction in the number of operations per circuit in terms of accuracy dependence (i.e., a reduction from $\tilde{O}(1/\epsilon)$ to $\tilde{O}(\log 1/\epsilon)$). This reduction in number of operations per circuit comes at the cost of an increase in circuit repetitions from $\tilde{O}(\log 1/\epsilon)$ to $\tilde{O}(1/\epsilon^2)$. With this method, the minimal number of operations per circuit is $\tilde{O}(1/\Delta)$, where Δ is a lower bound on the spectral gap of H . It is often the case that Δ is larger than ϵ , enabling a reduction in number of operations using this method [57]. Later, Ding and Lin [60] have established a similar result using an approach based on numerically fitting a parametrized curve to a set of estimated expectation values, which they refer to as quantum complex exponential least squares (QCELS). They have also shown that,

TABLE III. A scaling comparison of select ground-state energy-estimation methods for early fault-tolerant quantum computers: ϵ is the estimation error, γ is a lower bound on the ground-state overlap (i.e., $|\langle \psi | gs \rangle|$), Δ is a lower bound on the gap (i.e., $E_1 - E_0$), and α and δ are parameters that can be chosen to balance the number of operations per circuit and run time.

Ground-state energy-estimation method	Operations per circuit	Run time
Fourier filtering [66]	$\tilde{O}(\epsilon^{-1})$	$\tilde{O}(\gamma^{-4}\epsilon^{-1})$
Gaussian filter [57]	$\tilde{O}(\epsilon^{-\alpha}\Delta^{-1+\alpha})$	$\tilde{O}(\gamma^{-4}\epsilon^{-2+\alpha}\Delta^{1-\alpha})$
QCELS [60]	$\tilde{O}(\Delta^{-1}) + \delta/\epsilon$	$\tilde{O}(\gamma^{-4}\delta^{-(2+\mathcal{O}(1))}\Delta^{-1} + \delta/\epsilon)$
Rejection sampling [69]	$\tilde{O}(\epsilon^{-\alpha}\Delta^{-1+\alpha})$	$\tilde{O}(\gamma^{-2}\epsilon^{-2+\alpha}\Delta^{1-\alpha})$

assuming that the ground-state overlap $|\langle \psi | gs \rangle| = \gamma$ is sufficiently close to 1, the circuit depth of energy estimation can be made arbitrarily small while still retaining the Heisenberg-limit scaling.

These two methods that have enabled a reduction in the number of operations per circuit [57,60] have required a run time and number of circuit repetitions scaling as $O(1/\gamma^4)$. In contrast, for methods using more operations per circuit, a run-time scaling of $O(1/\gamma^2)$ [126] (and even $O(1/\gamma)$ [58]) has been shown to be possible. This has motivated the search for algorithms that could improve the run-time scaling with respect to overlap to $O(1/\gamma^2)$, while also using few operations per circuit. The first work to achieve this was Ref. [69]. This algorithm uses the quantum computer in a very different manner compared to previous approaches. To estimate the ground-state energy, a classical computer first generates a uniformly random sampling of energy values on an interval expected to contain the ground-state energy. Then, for each sample, a quantum circuit is designed such that a binary measurement outcome from the quantum computer is used to decide whether or not that sample should be accepted or rejected. The set of accepted samples are proven to be drawn from a Gaussian distribution centered about the ground-state energy and the mean of these samples will be close to this value. The number of operations per circuit can be tuned anywhere from $O(1/\epsilon)$ to $O(1/\Delta)$, where the consequence is a broadening of the Gaussian peak width (requiring more samples to achieve the same accuracy). Later, in Ref. [114], the authors have also developed a ground-state energy-estimation algorithm with $O(1/\Delta)$ operations per circuit and $O(1/\gamma^2)$ circuit repetitions based on a Gaussian-filter variant of QPE. Although this uses more ancilla qubits like QPE, Ref. [114] shows that the

number of operations per circuit is reduced compared to previous methods.

Another thread of research in ground-state energy estimation has drawn on methods from numerical linear algebra to classically postprocess quantum measurement-outcome data in a more efficient and robust manner [53,56,61,111,113,127]. These methods employ techniques such as filter diagonalization [53,56], Lanczos methods [61], and dynamic mode decomposition [111]. Some of these methods [61,127] have been shown to only require a number of operations per circuit scaling as $\tilde{O}(1/\Delta)$, similar to Ref. [57]. However, in the case of Ref. [61], the run-time upper bound has a scaling of $\tilde{O}(1/\Delta^2)$, compared to the $\tilde{O}(\Delta)$ run-time scaling of Ref. [57]. An important direction for future work will be to carry out empirical studies that give more realistic estimates for the run times, required operations per circuit, and robustness of these algorithms.

As mentioned above, all ground-state energy-estimation methods have a run time that depends on the overlap between the input trial state and the ground state $\langle \psi | gs \rangle = \gamma$. The run time of these ground-state energy-estimation algorithms can be improved by using a ground-state preparation method to improve this overlap before running the ground-state energy-estimation algorithm. For papers analyzing the interplay between ground-state energy estimation and state preparation, see Refs. [128–130]. An excellent overview of ground-state preparation algorithms is presented in Table II of Ref. [58]. In the regime of early fault-tolerant quantum computing, it may be advantageous to use ground-state preparation methods that reduce the number of operations per circuit. One such method has been proposed in Ref. [59], where an approximate Gaussian filter of varying width can be used to suppress high-energy states and boost the overlap with the ground state. More recently, in Ref. [115], the authors have introduced a novel ground-state preparation method based on Lindblad dynamics, engineering a process that has the ground state as the unique steady state.

Finally, we discuss another important thread of research for early fault-tolerant quantum computing: *robustness*. As shown in Sec. II B, p_C , the circuit error rate that the algorithm can tolerate, plays a role in determining the fault-tolerant overhead. An increase in the robustness (i.e., p_C) reduces the fault-tolerant overhead. A canonical reference on the robustness of quantum algorithms is Ref. [99], which introduced the robust phase-estimation algorithm. The authors have shown that their variant of QPE is able to tolerate a substantial circuit error rate of approximately 35%. Such robustness analysis is especially important when reduction of fault-tolerant overhead is essential, as in quantum algorithms suited for early fault-tolerant quantum computers.

One of the first works to analyze the robustness of a quantum algorithm in the EFTQC setting was Ref. [100]. Here, the simple robust phase-estimation algorithm

was introduced and its robustness was analyzed with respect to two different models of algorithmic noise (i.e., a model of how the measurement-outcome probabilities are impacted). For a discussion of these algorithmic noise models, see Sec. III B. A variant of the robust phase-estimation algorithm has been developed in Ref. [107] that enables circuit trading. Here, the robustness of the algorithm is analyzed with respect to the exponential-decay model (see Sec. III B). Most recently, in Ref. [117], the authors have developed an algorithm for ground-state energy estimation that is provably robust with respect to the exponential-decay model (see Eq. 16).

The works presented above represent a foundation for an increasingly important research direction: the development of quantum algorithms suited to the capabilities of finite-scalability quantum computers. They are built to reduce the logical qubit number, to enable a trade-off between the gate count and the sample cost, and they are robust to error in the circuit. All of these features contribute to reducing the burden placed on fault-tolerant operations and thus reducing fault-tolerant overheads. This helps us to run larger problem instances on earlier quantum computers or, in other words, to “extend the reach” of a finite-scalability quantum architecture. While we have focused on quantum algorithms for early fault-tolerant quantum computing, techniques such as error mitigation will also likely play an important role in supporting early fault-tolerant quantum computing. Similar to the way in which the above algorithms reduce gate counts at the cost of an increase in sample complexity, error-mitigation methods can be understood to increase the robustness of the quantum algorithm at the cost of an increase in sample complexity. In Sec. III B, we will make these concepts more clear through an example.

B. Example: Randomized Fourier estimation under finite scalability

Section II B ended with the question of how we might extend the reach of finite-scalability quantum computers. Section III A overviewed a host of quantum algorithms suited for addressing this question. In this section, we take one quantum algorithm from the previous section and quantitatively investigate its ability to extend the reach of a finite-scalability quantum computer for the task of phase estimation.

We use as our example the randomized-Fourier-estimation (RFE) algorithm as introduced in Ref. [100] and adapted for trading circuit repetitions for number of operations per circuit in Ref. [107]. The RFE algorithm solves the task of phase estimation introduced in Sec. II B. It is an alternative to the standard QPE algorithm [131] and related algorithms such as robust phase estimation (RPE) [99].

We consider the RFE algorithm to be a prototypical quantum algorithm suited for early fault-tolerant quantum computing given that it has the following features:

- (a) *Qubit conservation*: the (high-level) circuit conserves qubit count by using just one ancilla qubit.
- (b) *Circuit trading*: the number of operations per circuit is tuned by input parameter K , enabling a trade-off between this quantity and the required number of circuit repetitions.
- (c) *Robustness*: the algorithm is robust to circuit error and this robustness can be understood in terms of a signal corrupted by a noise floor.

As we will show, and like many of the other EFTQC algorithms introduced in Sec. III A, these features equip the algorithm to accommodate limited scalability in the early fault-tolerant quantum computing regime. Furthermore, RFE is very simple, helping to facilitate discussion of these algorithmic concepts relevant to early fault-tolerant quantum computing.

We will briefly review RFE and then investigate how trading circuit repetitions for decrease of operations and robustness to error help to increase the problem-instance size (i.e., the “reach”) that can be solved with a finite-scalability architecture.

1. Introduction to RFE

The RFE algorithm relies on the Hadamard test circuit (as depicted in Fig. 4). Each Hadamard test circuit is parametrized by the circuit depth (k) and a phase parameter (ϕ). The output-measurement probabilities correspond to an oscillatory function that encodes θ :

$$\Pr(z|\theta; k, \phi) = \frac{1}{2}(1 + z\text{Re}(e^{i\phi} \langle \psi | U^k | \psi \rangle)) \quad (11)$$

$$= \frac{1}{2}(1 + z \cos(k\theta + \phi)). \quad (12)$$

It is convenient to view the expected value of z , which is $g(k) = \cos(k\theta + \phi)$, as the true signal encoding θ . The phase θ is then estimated from measurement-outcome data in a manner similar to estimating the frequency of a noisy estimate of $g(k)$. The parameters k and ϕ are chosen uniformly randomly in each sample, with $k \in [0, K - 1]$ and ϕ ranging between 0 and 2π . Each measurement outcome z obtained from the circuit is used to form an unbiased estimator $\hat{f}_j = 2ze^{-i2\pi kj/J} e^{-i\phi}$ of the discrete Fourier transform of the signal $g(k)$, where J is an algorithm parameter that sets the grid size of the Fourier spectrum. The estimate of the Fourier signal can be made more accurate by taking multiple samples and averaging them:

$$\hat{f}_j = \frac{1}{M} \sum_{i=1}^M \hat{f}_j^{(i)}. \quad (13)$$

By accumulating enough measurement outcomes, one can estimate θ accurately (i.e., within ϵ) with high probability (i.e., less than $1 - \delta$) by locating the tallest peak (or point of largest magnitude) in the estimate of the discrete Fourier transform,

$$\hat{\theta} = \frac{2\pi}{J} \operatorname{argmax}_j \frac{1}{M} \sum_{i=1}^M \hat{f}_j^{(i)}. \quad (14)$$

The accuracy of the algorithm is limited by parameter J , which is set to ensure that the Fourier resolution matches the desired accuracy.

Algorithm parameters:

J	(sets the Fourier-domain grid spacing)
K	(sets the maximum number of c - U per circuit)
M	(sets the number of circuit repetitions, i.e., samples)

Error and confidence requirements:

$\epsilon = 2\pi/J$	(ensures that the θ -adjacent discrete frequencies are accurate)
$\delta = 8J \exp(-M/W(K, J, \lambda))$	(ensures that enough samples are taken, given K, J , and λ ; see below)

Operations per circuit:

$\mathbb{E}G_C = \frac{K-1}{2} G_U$	(expected value, with the maximum being $(K-1)G_U$)
-------------------------------------	--

Circuit repetitions:

$M = W(K, J, \lambda) \log(\frac{16\pi}{\delta\epsilon})$	(number of samples needed; for definition of $W(K, J, \lambda)$, see Ref. [107])
---	---

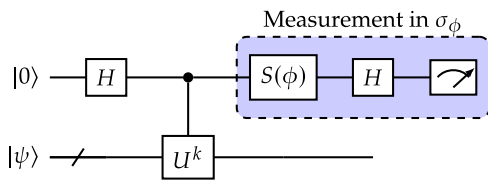


FIG. 4. An archetypal circuit template used by many EFTQC algorithms. The measurement-outcome probabilities depend on $|\psi\rangle$ and U as $\Pr(\pm 1|k, \phi) = \frac{1}{2}(1 \pm \cos(k\theta + \phi))$. Measurement outcomes can be processed to In the case of the randomized Fourier estimation (RFE) algorithm, the measurement outcomes encode the The parameter k is uniformly randomly chosen among $\{0, \dots, K-1\}$ for each circuit repetition. K then controls the maximal circuit depth and is used to reduce the number of operations per circuit. We define $S(\phi) = [[1, 0], [0, \exp(i * \phi)]]$ and the elements in the blue box can be collectively interpreted as a measurement with respect to the observable $\sigma_\phi = \cos(\phi)\sigma_x - \sin(\phi)\sigma_y$, where σ_x and σ_y are the conventional Pauli operators.

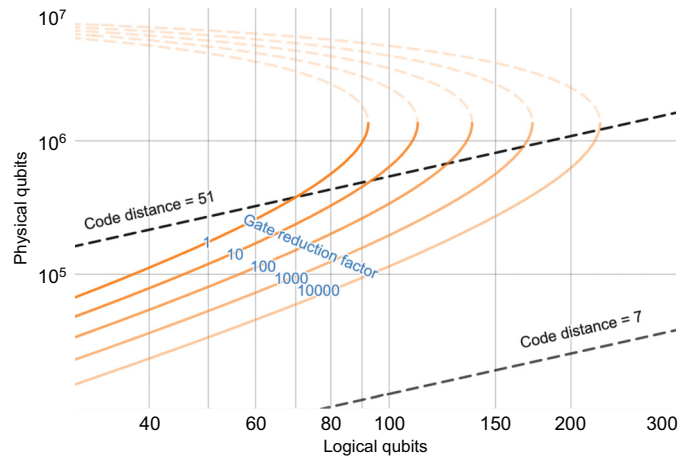


FIG. 5. This plot shows that under the scalability model, the EFTQC algorithm randomized Fourier estimation (RFE) can extend the reach of the quantum computation from 90 logical qubits to over 200 logical qubits. This is achieved by either reducing the number of c - U used per circuit or increasing the tolerable circuit error rate p_C in the RFE algorithm. Both of these reduce the burden factor $A\alpha/\epsilon p_C$ appearing in Eq. (7). This increase in the “reach” of the quantum computer comes at the cost of an increase in the run time (roughly by the burden factor), which is a combination of the decrease in time per circuit and the increase in the number of circuit repetitions. Here, we take the scalability to be $s = 3.5$ (this puts us in the millions of physical qubits likely regime for first useful quantum computation) with $p_0 = 10^{-4}$, which implies that the optimal number of physical qubits is $Q_{\text{phys}}^{\text{opt}} = 1/e^2 (p_{\text{th}}/p_0)^s \approx 1.35 \times 10^6$. An editable version of the plot can be accessed at Ref. [132].

2. Circuit trading

We now describe how this algorithm is able to trade the number of operations per circuit for circuit repetitions. The maximum number of operations per circuit (in expectation) is $(K-1)G_U$, where G_U is the number of operations in a single c - U . In the QPE algorithm, $1/\epsilon$ calls are made to c - U , corresponding to setting $K \approx 1/\epsilon$. In RFE, we can reduce the number of operations per circuit by setting K to any value less than $1/\epsilon$. This reduction in K reduces the burden factor in Eqs. (7) and (10) proportionally. In Fig. 5, we show how varying reductions in the burden factor lead to an increase in the problem size that RFE can accommodate. Equation (10) predicts that this increase in problem size grows as $O(1/\ln^2(B))$ with burden factor B . For the specific example considered, the largest problem instance can be increased from 90 to over 200 by decreasing K by a factor of 100 000.

As mentioned previously, circuit trading means that a decrease in operations per circuit comes at the cost of an increase in the number of circuit repetitions. This trade-off can be understood as follows. Decreasing K causes the *width* of the peak in the discrete Fourier spectrum to increase. With the spectrum being more flat near the peak,

smaller amounts of noise in the signal are able to shift the peak location more than ϵ (leading to algorithm failure). This statistical sampling noise must then be reduced by taking more samples. The analytic relationship is given in the appendix of Ref. [107]. This describes the nature of the trade-off between operations per circuit and circuit repetitions.

3. Robustness

The RFE algorithm has been analyzed in previous work with respect to three different algorithmic noise models: adversarial noise and Gaussian noise [100] and exponential-decay noise [107]. We give brief explanations of how the Gaussian noise and the exponential-decay noise impact the algorithm performance and thus explain the robustness of the RFE algorithm to a particular model of noise. In Ref. [100], the Gaussian-noise model is analyzed, wherein it is assumed that for each circuit (labeled by k), the output probability has been corrupted by a small perturbation drawn from a Gaussian distribution,

$$\Pr(z|\theta; k, \phi) = \frac{1}{2}(1 + z(\cos(k\theta + \phi) + \eta_k)), \quad (15)$$

where each η_k has been drawn from a Gaussian distribution with mean zero and standard deviation σ . How does this impact the performance of the algorithm? The η_k can be understood to corrupt the expected value of z (i.e., the signal $g(k)$). This impacts the Fourier spectrum by adding a “noise floor” related to the Fourier transform of the η_k . The algorithm can still succeed as long as this noise floor does not shift the location of the peak by more than ϵ . In Ref. [100], the authors have proved that if σ is below a certain quantity (dependent on ϵ and δ), then the algorithm can succeed with more than $1 - \delta$ probability.

In Ref. [107], the exponential-decay model is derived from a lower-level noise model. The exponential-decay model assumes that the likelihood function now includes a factor that decreases exponentially in k ,

$$\Pr(z|\theta; k, \phi) = \frac{1}{2}(1 + ze^{-k\lambda} \cos(k\theta + \phi)), \quad (16)$$

with decay parameter λ . Experiments [119,120] show that this model is accurate for small systems. This exponential-decay factor causes the expected value of z (i.e., the underlying signal $g(k)$) to attenuate as k is increased. In the Fourier domain, this attenuation translates into an attenuation of the peak (see the desmos plot at Ref. [133]). As with the peak broadening due to reducing K , a smaller amount of statistical noise is sufficient to shift the location of the estimated peak more than ϵ . Accordingly, more samples must be taken to sufficiently reduce this statistical noise.

Under the assumption that the exponential-decay model holds exactly, Ref. [107] shows that with arbitrarily large

decay parameter λ , the algorithm can generate an ϵ accurate estimate with probability greater than $1 - \delta$. In other words, the algorithm can be made arbitrarily robust. The reason is that the exponential-decay error does not shift the location of the peak in the Fourier spectrum of the expected signal. This increase in robustness translates into a decrease in the burden: allowing the circuit error rate p_C to increase toward 1 increases the allowed logical error rate p_L , decreasing the burden factor.

Consider a reduction in the burden on account of an increase in the tolerable circuit error rate p_C , which quantifies the robustness of the algorithm. Note that in the case in which p_C is close to 1, a better approximation than the union bound can be used to replace p_C with $\ln(1/(1 - p_C))$, which grows to infinity as $p_C \rightarrow \infty$. We remark that in the case of the exponential-decay model, the circuit error rate is $p_C = 1 - e^{-k\lambda}$, which leads to $\ln(1/(1 - p_C)) = k\lambda$. Therefore, as we allow for an increase in λ , the burden factor is reduced proportionally (where we keep in mind that, for small values of p_C , the burden factor scales proportionally to it).

We have previously discussed Fig. 5 in the context of circuit trading. This figure can also be used to demonstrate the impact of increased robustness. Considering an increase in p_C to be the cause of the burden-factor reduction, in Fig. 5, we show how the reach of the quantum computer is increased accordingly. As with circuit trading, there is a price paid for this extended reach of the quantum computer: for the RFE algorithm, Ref. [107] shows that the run time grows exponentially in λ for $\lambda \geq 1/2$ (where K is set to its minimum value of 2). Therefore, in practice, there may be an upper limit to the degree of robustness, beyond which the run time becomes too large to be practical. This is an issue that many error-mitigation techniques face [134]. This similarity may not be surprising in that the way in which RFE accommodates error is a type of error mitigation.

In practice, the exponential-decay model is not exact. Instead, we expect that in any given device and compilation of c - U , the likelihood function will include some deviation (possibly varying over time) from the exponential-decay model likelihoods. While in the exact exponential-decay model the Fourier peak location is unchanged, allowing deviations from this model can shift the location of the peak. This sets a lower limit to the achievable accuracy ϵ , a feature that is found in the bounded adversarial-noise model and the Gaussian-noise model of Ref. [100].

We have demonstrated how the RFE algorithm, as an archetypal EFTQC algorithm, enables a reduction in the burden placed on the fault-tolerant protocols. In Fig. 5, we demonstrate how larger problem-instance sizes can be accommodated by either reducing the number of operations per circuit (decreasing K) or by increasing the robustness of the algorithm (increasing p_C). This is because

the burden factor $A\alpha/p_C$ incorporates both of these quantities. For both examples of reducing the burden factor, there is an increase in the run time of the algorithm. Although the RFE algorithm enables parallelizing the circuit repetitions over multiple quantum computers to reduce run time, the run time is expected to be a bottleneck for many applications. Therefore, the run-time costs of reducing the fault-tolerance burden must be carefully considered (for a quantitative account of such run-time costs for RFE, see Ref. [107]). We leave a thorough investigation of the run-time costs of decreasing the fault-tolerance burden for EFTQC algorithms to future work.

IV. DISCUSSION AND OUTLOOK

In this perspective, we have investigated the regime between NISQ and FTQC, which is referred to as “early fault-tolerant quantum computing.” To understand the prospects for utility in this regime, we have proposed a simple computational model to quantitatively capture the performance of quantum architectures within these three regimes. The *scalability* model characterizes the ability of a quantum hardware vendor to provide systems with low physical error rates as the requested number of physical qubits is increased. This differs from previous approaches that assume a scale-independent performance for their quantum architectures [70,71,86]. We have demonstrated that the QPE algorithm [62] compiled to the surface code [105] has a limit on the problem size that can be accommodated by a vendor with finite scalability, according to our model. Unsurprisingly, this is due to scale-dependent error rates [Eq. (1)] combined with the diminishing returns of fault-tolerant protocols as the error rates of the device approach the numerically estimated threshold value [75]. Next, we have shown that by using an algorithm suited to finite scalability (the RFE algorithm [100]), when granted the same scalability, the problem size limit can be extended from around 90 qubits (for QPE) to around 130 qubits (using the same number of physical qubits). This comes at the cost of a roughly factor-of-100 increase in run time.

The scalability model has enabled us to quantitatively discuss the transition from NISQ to EFTQC to FTQC. This transition is characterized by the waning of scale-dependent physical error rates (see Eq. (1)). At the end of Sec. II A, we have described how the nature of the transition from the regime of NISQ to EFTQC is difficult to predict; future advances might allow for implementing certain fault-tolerant components far sooner than current methods would enable. However, we have mentioned some of the technical considerations that might govern the transition and, accordingly, we depict this transition in Fig. 1 to occur through the range of $Q_{\text{phys}}^{\text{max}}$ being 100–10 000. Regarding the transition from EFTQC to FTQC, we have described in Sec. II A how each regime might be characterized by different bottlenecks; EFTQC is characterized by the largest

solvable problem instances being bottlenecked by the number of available physical qubits (or, better, $Q_{\text{phys}}^{\text{opt}}$), whereas FTQC is characterized by the largest solvable problem instances being bottlenecked by the run time. Accordingly, we explain how this transition might occur in the range of $Q_{\text{phys}}^{\text{max}}$ being 10^6 – 10^8 .

Different factors, such as hardware, algorithmic, and fault-tolerance advances, play a dominant role in characterizing the EFTQC regime. The recent work in Ref. [43] provides evidence for the utility of noisy quantum devices in the pre-fault-tolerant era and emphasizes the role of hardware advances to achieve this. Moreover, many works have highlighted the importance that quantum algorithm development has in leveraging the capabilities of the quantum devices to their maximum potential [52,54,119,135]. Recent works have also explored the effect of noise in the performance of quantum algorithms and highlighted the need to use QEC prudently [107,118,126,136,137]. This work is a first attempt to incorporate all the aforementioned factors (hardware, algorithmic, and fault-tolerance advances) in order to validate the assumption that there is a meaningful regime of early fault-tolerant quantum computing methods, which is usually assumed in papers on the subject [57,60,66]. What remains to be determined is how rapidly quantum hardware will progress through this regime; or, in other words, it remains to determine how the scalability of quantum hardware vendors will increase over time.

To put these results into context, recent resource estimates on a variety of molecules relevant to Li-ion electrolyte chemistry [138] show that above 100 logical qubits would be necessary to tackle such systems. This indicates that extending the reach of the problem size limit from 90 logical qubits to over 130 with the framework discussed here might have interesting implications, i.e., allowing us to study problems of interest before the realization of the FTQC regime. Our results suggest that the EFTQC regime could exist in a meaningful way, i.e., using the same quantum resources compared to FTQC (number of physical qubits and scalability model), while affording the use of a larger number of logical qubits (Fig. 5).

This work has explored the usefulness of the EFTQC regime for a specific quantum algorithm and QEC model, namely, the RFE [100] and the surface code [105]. The underlying methodology, however, can be easily extended to other algorithms and fault-tolerant protocols while using the suggested or alternative scalability models. Although the proposed model of scalability is quite general, we do not expect it to perfectly fit the scalability profile of vendors over many orders of magnitude. But, we anticipate that it could capture the qualitative behavior over at least a few orders of magnitude. Moreover, we have shown in Appendix B that even when a more optimistic model is used (specifically a logarithmic model), the qualitative

finding remains: there is an upper limit on the size of the quantum computation.

Future work could explore other models of scalability that might, e.g., be given directly by the hardware provider and accommodate the features of the architecture as it is scaled. Another interesting direction is to adapt the scalability model to address the interplay between quantum error mitigation (including error detection) and quantum error correction [139,140], which will help drive the transition from NISQ to EFTQC. Moreover, the proposed framework could be applied to other combinations of algorithms and quantum error-correcting codes and be used to examine the utility of the EFTQC regime for other potential application fields of quantum computing.

Our work provides evidence for the utility of the EFTQC regime within a framework that includes crucial factors of quantum computing, such as hardware, algorithm, and fault-tolerance advances. To incorporate the hardware advances, we have introduced a simple scalability model to capture the performance of devices that are continually improving. As it is yet unclear how exactly quantum devices will scale up to incorporate millions or billions of physical qubits [42], the proposed model of scalability is just a first attempt to bridge the gap between NISQ and FTQC. Future works in these directions could help move beyond the NISQ-FTQC dichotomy and further explore how EFTQC might deliver practical quantum advantage at scale.

ACKNOWLEDGMENTS

We extend our gratitude to Lin Lin and Zhiyan Ding for their invaluable discussions and for sharing their insights on early fault-tolerant quantum computing. Their substantial feedback has significantly enhanced the clarity and presentation of our paper. We thank Shangjie Guo for a careful reading of the paper, improving the presentation. We also thank Daniel Stilck França and Guoming Wang for many inspiring conversations on the topic of early fault-tolerant quantum computing, with special thanks to Daniel Stilck França for contributing material to an earlier version of the paper. Our appreciation also goes to the team of Zapata scientists whose thoughtful suggestions shaped the presentation of this work. We are particularly indebted to Max Radin, Nicole Bellonzi, Vlad Vargas-Calderón, Shima Bab Hadiashar, Matt Kowalsky, Yudong Cao, and others for their constructive critiques and expertise that has greatly contributed to the refinement of our ideas and methodologies presented in this paper. Finally, we thank Alán Aspuru-Guzik, Alejandro Perdomo-Ortiz, Aram Harrow, and Will Oliver for early discussions on the topic of early fault-tolerant quantum computing that helped motivate this work. This work was performed while K.G. was a research intern at Zapata AI, Inc. Part of this research was performed while K.G. was visiting the Institute for Pure

and Applied Mathematics (IPAM), which is supported by the National Science Foundation (Grant No. DMS-1925919). K.G. acknowledges support from the European Union (EU) Horizon 2020 research and innovation program under the Marie Skłodowska-Curie Grant Agreement No. 847517. The ICFO group acknowledges support from the European Research Council (ERC) Advanced Grant (AdG) “NOvel Quantum simulators—connectIng Areas” (NOQIA); Ministerio de Ciencia y Innovation Agencia Estatal de Investigaciones (PGC2018-097027-B-I00/10.13039/501100011033, CEX2019-000910-S/10.13039/501100011033, Plan National FIDEUA PID2019-106901GB-I00, FPI, QuantERA Magnetic-Atom Quantum Simulator (MAQS) PCI2019-111828-2, QuantERA DYNAMITE PCI2022-132919, Proyectos de I+D+I “Retos Colaboración” QUSPIN RTC2019-007196-7); MICIIN, with funding from the EU NextGenerationEU (PRTR-C17.I1) and by Generalitat de Catalunya; Fundació Cellex; Fundació Mir-Puig; Generalitat de Catalunya (European Social Fund FEDER and CERCA program, Agencia de Gestión de Ayudas Universitarias y de Investigación (AGAUR) Grant No. 2021 SGR 01452, QuantumCAT U16-011424, cofunded by the European Regional Development Fund (ERDF) Operational Program of Catalonia 2014–2020); Barcelona Supercomputing Center MareNostrum (FI-2023-1-0013); EU (PASQuanS2.1, 101113690); EU Horizon 2020 FET Open Optical Topologic Logic (OPTologic) (Grant No. 899794); EU Horizon Europe Program (Grant Agreement No. 101080086—NeQST), National Science Center, Poland (Symfonia Grant No. 2016/20/W/ST4/00314); ICFO Internal “QuantumGaudi” project; EU Horizon 2020 research and innovation program under the Marie-Skłodowska-Curie Grant Agreements No. 101029393 (STREDCH) and No. 847648 (“La Caixa” Junior Leaders fellowships ID100010434: LCF/BQ/PI19/11690013, LCF/BQ/PI20/11760031, LCF/BQ/PR20/11770012, LCF/BQ/PR21/11840013). The views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union, the European Commission, European Climate, the Infrastructure and Environment Executive Agency (CINEA), or any other funding authority. Neither the European Union nor any funding authority can be held responsible for them.

APPENDIX A: SCALABILITY OF TODAY’S DEVICES

Here, we estimate the scalability of today’s quantum devices. To this end, we have collected data from two IBM devices available on the cloud, namely, *lagos* and *brisbane*. We have collected the CNOT error rates at 10:00 a.m. each day over 30 d. In Fig. 6, we plot the mean and standard deviation of the worst-case CNOT error rates for the studied devices. We then make a power-law fit of the data to

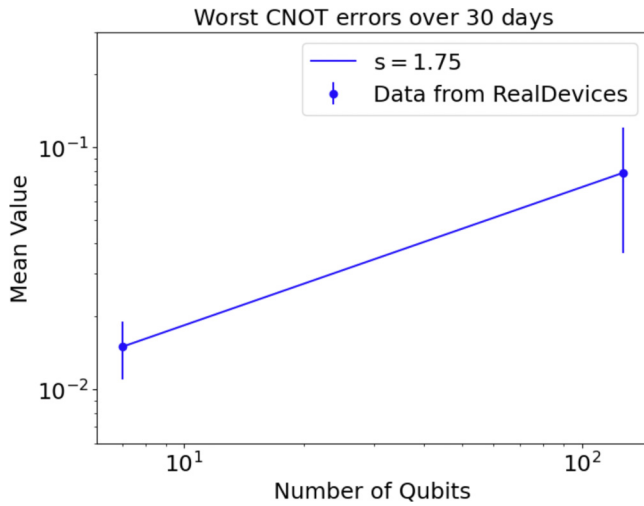


FIG. 6. The worst two-qubit gate error of two IBM quantum devices on the cloud as a function of the number of qubits. The power-law fit (blue line) suggests that today’s scalability is $s = 1.75$ and $p_0 = 0.005$.

estimate the p_0 and s as introduced in Eq. (1). We find that $p_0 = 0.005$ and $s = 1.75$, which we refer to as today’s scalability. In Fig. 1, we show this point to lie in the NISQ regime, despite p_0 being below threshold.

It is very hard to guess what the right scalability model of future devices will be and different architectures will most probably have very different profiles. However, the scope of this perspective paper is to raise awareness and spawn more research on the topic, especially by collaborating with hardware designers who will provide more realistic scalability models. Even if the scalability model turns out to be completely wrong, the quantitative results discussed in the main text will be different but the qualitative results will remain the same—predicting that a maximum size limit will emerge due to practical limitations of the scalability profiles. The goal of this work is to show that resource-estimation tools could help make tentative assumptions and evaluate their consequences.

APPENDIX B: LOGARITHMIC SCALABILITY MODEL

In this appendix, we investigate the implications of a more optimistic scalability model. Instead of the power-law model of Eq. (1), we consider a logarithmic model for the scalability profile,

$$p_{\text{phys}}(Q_{\text{phys}}; \mathcal{V}) = p_0 \left(1 + \frac{1}{\sigma} \ln(Q_{\text{phys}}) \right), \quad (\text{B1})$$

where σ is the scalability parameter analogous to s in Eq. (1). The parametrization is chosen such that at $Q_{\text{phys}} = 1$, the function value and slope match those of Eq. (1). With

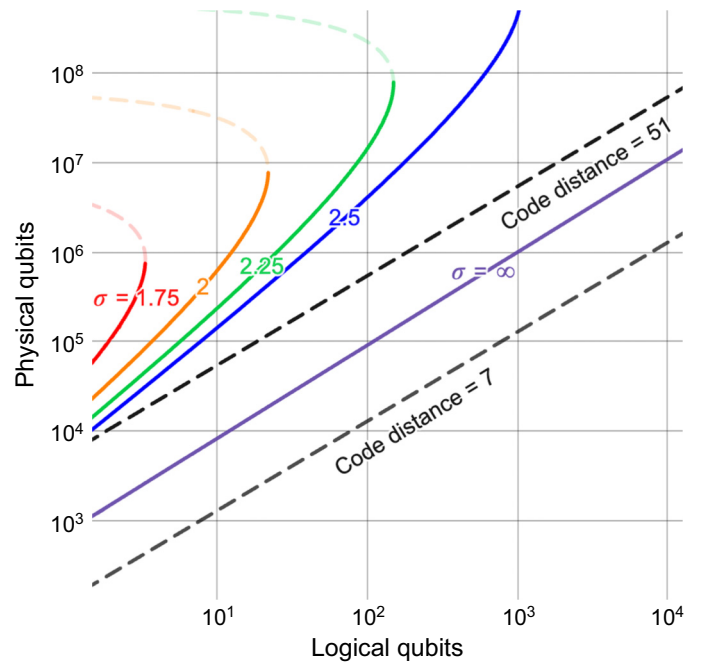


FIG. 7. The QPE resource overhead under the logarithmic scalability model of Eq. (B1). Similar to the power-law model shown in Fig. 3, the logarithmic scalability model also predicts that for each finite value of scalability parameter σ , there is a maximum problem-instance size that can be accommodated by the architecture. However, the logarithmic scalability model is more optimistic in that for the same base error rate $p_{\text{phys}}(Q_{\text{phys}} = 1) = p_0$ and for the same (logarithmic) slope at $Q_{\text{phys}} = 1$, the maximum problem-instance size is far larger for the logarithmic scalability model. Thus, in order to observe the limited problem-instance size in the range of 10–10 000 logical qubits, we use the larger base error rate of $p_0 = 0.001$ compared to the $p_0 = 0.0001$ used in Fig. 3. An editable version of the plot can be accessed at Ref. [141].

this alternative scalability model, the physical-qubit number at which the physical error rate exceeds the threshold value is now

$$Q_{\text{phys}}^{\text{max}} = \exp\left(\sigma \frac{p_0 - p_{\text{th}}}{p_{\text{th}}}\right), \quad (\text{B2})$$

which, compared to the power-law model, grows exponentially in the gap between p_0 and p_{th} .

[1] R. P. Feynman, Simulating physics with computers, *Int. J. Theor. Phys.* **21**, 467 (1982).
 [2] D. Deutsch and R. Jozsa, Rapid solution of problems by quantum computation, *Proc. R. Soc. Lond. Ser. A: Math. Phys. Sci.* **439**, 553 (1992).
 [3] L. K. Grover, Quantum mechanics helps in searching for a needle in a haystack, *Phys. Rev. Lett.* **79**, 325 (1997).

- [4] E. Bernstein and U. Vazirani, in *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, New York, NY, United States, 1993), p. 11.
- [5] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM J. Comput.* **26**, 1484 (1997).
- [6] P. W. Shor, Scheme for reducing decoherence in quantum computer memory, *Phys. Rev. A* **52**, R2493 (1995).
- [7] A. M. Steane, Active stabilization, quantum computation, and quantum state synthesis, *Phys. Rev. Lett.* **78**, 2252 (1997).
- [8] R. Laflamme, C. Miquel, J. Pablo Paz, and W. H. Zurek, Perfect quantum error correcting code, *Phys. Rev. Lett.* **77**, 198 (1996).
- [9] E. Knill and R. Laflamme, Concatenated quantum codes, [ArXiv:quant-ph/9608012](https://arxiv.org/abs/quant-ph/9608012).
- [10] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, Quantum error correction and orthogonal geometry, *Phys. Rev. Lett.* **78**, 405 (1997).
- [11] A further requirement is that realistic assumptions be made on the noise model.
- [12] P. W. Shor, in *Proceedings of 37th Conference on Foundations of Computer Science* (IEEE, Burlington, VT, USA, 1996), p. 56.
- [13] D. Aharonov, A. Kitaev, and J. Preskill, Fault-tolerant quantum computation with long-range correlated noise, *Phys. Rev. Lett.* **96**, 050504 (2006).
- [14] D. Aharonov and M. Ben-Or, Fault-tolerant quantum computation with constant error rate, *SIAM J. Comput.* **38**, 1207 (2008).
- [15] A. Yu. Kitaev, Fault-tolerant quantum computation by anyons, *Ann. Phys. (NY)* **303**, 2 (2003).
- [16] P. Aliferis and D. W. Leung, Simple proof of fault tolerance in the graph-state model, *Phys. Rev. A* **73**, 032308 (2006).
- [17] P. Aliferis and J. Preskill, Fault-tolerant quantum computation against biased noise, *Phys. Rev. A: At., Mol., Opt. Phys.* **78**, 052331 (2008).
- [18] P. Aliferis, D. Gottesman, and J. Preskill, Accuracy threshold for postselected quantum computation, *Quantum Inf. Comput.* **8**, 181 (2008).
- [19] P. Aliferis, D. Gottesman, and J. Preskill, Quantum accuracy threshold for concatenated distance-3 code, *Quantum Inf. Comput.* **6**, 97 (2006).
- [20] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Surface codes: Towards practical large-scale quantum computation, *Phys. Rev. A* **86**, 032324 (2012).
- [21] A. A. Kovalev and L. P. Pryadko, Fault tolerance of quantum low-density parity check codes with sublinear distance scaling, *Phys. Rev. A: At., Mol., Opt. Phys.* **87**, 020304 (2013).
- [22] N. P. Breuckmann and J. N. Eberhardt, Quantum low-density parity-check codes, *PRX Quantum* **2**, 040101 (2021).
- [23] L. Z. Cohen, I. H. Kim, S. D. Bartlett, and B. J. Brown, Low-overhead fault-tolerant quantum computing using long-range connectivity, *Sci. Adv.* **8**, 1717 (2022).
- [24] K. Wintersperger, F. Dommert, T. Ehmer, A. Houshanov, J. Klepsch, W. Mauerer, G. Reuber, T. Strohm, M. Yin, and S. Luber, Neutral atom quantum computing hardware: Performance and end-user perspective, *EPJ Quantum Technol.* **10**, 1 (2023).
- [25] X. F. Shi, Quantum logic and entanglement by neutral Rydberg atoms: Methods and fidelity, *Quantum Sci. Technol.* **7**, 023002 (2022).
- [26] S. Debnath, N. M. Linke, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, Demonstration of a small programmable quantum computer with atomic qubits, *Nature* **536**, 63 (2016).
- [27] D. Hanneke, J. P. Home, J. D. Jost, J. M. Amini, D. Leibfried, and D. J. Wineland, Realization of a programmable two-qubit quantum processor, *Nat. Phys.* **6**, 13 (2009).
- [28] C. Monroe, D. M. Meekhof, B. E. King, W. M. Itano, and D. J. Wineland, Demonstration of a fundamental quantum logic gate, *Phys. Rev. Lett.* **75**, 4714 (1995).
- [29] S. Bao, S. Kleer, R. Wang, and A. Rahmani, Optimal control of superconducting qmon qubits using Pontryagin's minimum principle: Preparing a maximally entangled state with singular bang-bang protocols, *Phys. Rev. A* **97**, 062343 (2018).
- [30] Y. Chen *et al.*, Qubit architecture with high coherence and fast tunable coupling, *Phys. Rev. Lett.* **113**, 220502 (2014).
- [31] D. Bluvstein, S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kalinowski, D. Hangleiter, J. P. Bonilla Ataides, N. Maskara, I. Cong, X. Gao, P. Sales Rodriguez, T. Karolyshyn, G. Semeghini, M. J. Gullans, M. Greiner, V. Vuletić, and M. D. Lukin, Logical quantum processor based on reconfigurable atom arrays, *Nature* **626**, 58 (2023).
- [32] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, *Nat. Commun.* **5**, 4213 (2014).
- [33] J.-G. Liu and L. Wang, Differentiable learning of quantum circuit Born machines, *Phys. Rev. A* **98**, 062324 (2018).
- [34] P.-L. Dallaire-Demers and N. Killoran, Quantum generative adversarial networks, *Phys. Rev. A* **98**, 012324 (2018).
- [35] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, [ArXiv:1411.4028](https://arxiv.org/abs/1411.4028).
- [36] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. GSL Brandao, D. A. Buell, *et al.*, Quantum supremacy using a programmable superconducting processor, *Nature* **574**, 505 (2019).
- [37] K. Temme, S. Bravyi, and J. M. Gambetta, Error mitigation for short-depth quantum circuits, *Phys. Rev. Lett.* **119**, 180509 (2017).
- [38] P. Czarnik, A. Arrasmith, P. J. Coles, and L. Cincio, Error mitigation with Clifford quantum-circuit data, *Quantum* **5**, 592 (2021).
- [39] T. Giurgica-Tiron, Y. Hindy, R. LaRose, A. Mari, and W. J. Zeng, *Digital zero noise extrapolation for quantum error mitigation* (IEEE, Denver, CO, USA, 2020), p. 306.
- [40] W. J. Huggins, S. McArdle, T. E. O'Brien, J. Lee, N. C. Rubin, S. Boixo, K. Birgitta Whaley, R. Babbush, and J. R. McClean, Virtual distillation for quantum error mitigation, *Phys. Rev. X* **11**, 041036 (2021).

- [41] E. van den Berg, Z. K. Mineev, A. Kandala, and K. Temme, Probabilistic error cancellation with sparse Pauli-Lindblad models on noisy quantum processors, *Nat. Phys.* **19**, 1116 (2023).
- [42] J. Preskill, Quantum computing in the NISQ era and beyond, *Quantum* **2**, 79 (2018).
- [43] Y. Kim, A. Eddins, S. Anand, K. Xuan Wei, E. van den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme, and A. Kandala, Evidence for the utility of quantum computing before fault tolerance, *Nature* **618**, 500 (2023).
- [44] K. Kechedzhi, S. V. Isakov, S. Mandrà, B. Villalonga, X. Mi, S. Boixo, and V. Smelyanskiy, Effective quantum volume, fidelity and computational cost of noisy quantum processing experiments, *Future Gener. Comput. Syst.* **153**, 431 (2024).
- [45] J. Tindall, M. Fishman, E. Miles Stoudenmire, and D. Sels, Efficient tensor network simulation of IBM's Eagle kicked Ising experiment, *PRX Quantum* **5**, 010308 (2024).
- [46] T. Begušić and G. K. Chan, Fast classical simulation of evidence for the utility of quantum computing before fault tolerance, *ArXiv:2306.16372*.
- [47] C. Piveteau and D. Sutter, Circuit knitting with classical communication, *IEEE Trans. Inf. Theory* **70**, 2734 (2023).
- [48] J. F. Gonthier, M. D. Radin, C. Buda, E. J. Duskocil, C. M. Abuan, and J. Romero, Measurements as a roadblock to near-term practical quantum advantage in chemistry: Resource analysis, *Phys. Rev. Res.* **4**, 033154 (2022).
- [49] E. Farhi and A. W. Harrow, Quantum supremacy through the quantum approximate optimization algorithm, *ArXiv:1602.07674*.
- [50] R. Shaydulin, C. Li, S. Chakrabarti, M. DeCross, D. Herman, N. Kumar, J. Larson, D. Lykov, P. Minssen, Y. Sun, *et al.*, Evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem, *ArXiv:2308.02342*.
- [51] D. Lykov, J. Wurtz, C. Poole, M. Saffman, T. Noel, and Y. Alexeev, Sampling frequency thresholds for the quantum advantage of the quantum approximate optimization algorithm, *npj Quantum Inf.* **9**, 1 (2023).
- [52] D. Wang, O. Higgott, and S. Brierley, Accelerated variational quantum eigensolver, *Phys. Rev. Lett.* **122**, 140504 (2019).
- [53] R. M. Parrish and P. L. McMahon, Quantum filter diagonalization: Quantum eigendecomposition without full quantum phase estimation, *ArXiv:1909.08925*.
- [54] G. Wang, D. E. Koh, P. D. Johnson, and Y. Cao, Minimizing estimation runtime on noisy quantum computers, *PRX Quantum* **2**, 010346 (2021).
- [55] T. Giurgica-Tiron, I. Kerenidis, F. Labib, A. Prakash, and W. Zeng, Low depth algorithms for quantum amplitude estimation, *Quantum* **6**, 745 (2022).
- [56] K. Klymko, C. Mejuto-Zaera, S. J. Cotton, F. Wudarski, M. Urbanek, D. Hait, M. Head-Gordon, K. B. Whaley, J. Moussa, N. Wiebe, *et al.*, Real-time evolution for ultra-compact Hamiltonian eigenstates on quantum hardware, *PRX Quantum* **3**, 020323 (2022).
- [57] G. Wang, D. S. França, R. Zhang, S. Zhu, and P. D. Johnson, Quantum algorithm for ground state energy estimation using circuit depth with exponentially improved dependence on precision, *Quantum* **7**, 1167 (2023).
- [58] Y. Dong, L. Lin, and Y. Tong, Ground-state preparation and energy estimation on early fault-tolerant quantum computers via quantum eigenvalue transformation of unitary matrices, *PRX Quantum* **3**, 040305 (2022).
- [59] G. Wang, S. Sim, and P. D. Johnson, State preparation boosters for early fault-tolerant quantum computation, *Quantum* **6**, 829 (2022).
- [60] Z. Ding and L. Lin, Even shorter quantum circuit for phase estimation on early fault-tolerant quantum computers with applications to ground-state energy estimation, *ArXiv:2211.11973*.
- [61] W. Kirby, M. Motta, and A. Mezzacapo, Exact and efficient Lanczos method on a quantum computer, *Quantum* **7**, 1018 (2023).
- [62] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, Quantum algorithms revisited, *Proc. R. Soc. Lond. Ser. A: Math. Phys. Eng. Sci.* **454**, 339 (1998).
- [63] <https://www.desmos.com/calculator/ui5ch0jdl9>
- [64] E. T. Campbell, Early fault-tolerant simulations of the Hubbard model, *Quantum Sci. Technol.* **7**, 015007 (2021).
- [65] R. Zhang, G. Wang, and P. Johnson, Computing ground state properties with early fault-tolerant quantum computers, *Quantum* **6**, 761 (2021).
- [66] L. Lin and Y. Tong, Heisenberg-limited ground-state energy estimation for early fault-tolerant quantum computers, *PRX Quantum* **3**, 010318 (2022).
- [67] K. Wan, M. Berta, and E. T. Campbell, Randomized quantum algorithm for statistical phase estimation, *Phys. Rev. Lett.* **129**, 030503 (2022).
- [68] Z. Ding and L. Lin, Even shorter quantum circuit for phase estimation on early fault-tolerant quantum computers with applications to ground-state energy estimation, *PRX Quantum* **4**, 020331 (2023).
- [69] G. Wang, D. Stilck França, G. Rendon, and P. D. Johnson, Faster ground state energy estimation on early fault-tolerant quantum computers via rejection sampling, *ArXiv:2304.09827*.
- [70] I. H. Kim, Y.-H. Liu, S. Pallister, W. Pol, S. Roberts, and E. Lee, Fault-tolerant resource estimate for quantum chemical simulations: Case study on Li-ion battery electrolyte molecules, *Phys. Rev. Res.* **4**, 023019 (2022).
- [71] J. J. Goings, A. White, J. Lee, C. S. Tautermann, M. Degroote, C. Gidney, T. Shiozaki, R. Babbush, and N. C. Rubin, Reliably assessing the electronic structure of cytochrome P450 on today's classical computers and tomorrow's quantum computers, *Proc. Natl. Acad. Sci.* **119**, e2203533119 (2022).
- [72] C. Zalka, Threshold estimate for fault tolerant quantum computation, *ArXiv:quant-ph/9612028*.
- [73] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Topological quantum memory, *J. Math. Phys.* **43**, 4452 (2002).
- [74] Google Quantum AI, Suppressing quantum errors by scaling a surface code logical qubit, *Nature* **614**, 676 (2023).
- [75] D. S. Wang, A. G. Fowler, and L. C. L. Hollenberg, Surface code quantum computing with error rates over 1%, *Phys. Rev. A* **83**, 020302 (2011).
- [76] A. G. Fowler, Proof of finite surface code threshold for matching, *Phys. Rev. Lett.* **109**, 180502 (2012).

- [77] In the surface code, the degree of resilience to error is controlled by the size of the two-dimensional grid of qubits used to encode each qubit. The minimal number of single-qubit errors needed to cause a logical error, or the code distance, is the diameter of the two-dimensional grid of qubits.
- [78] X. Xue, M. Russ, N. Samkharadze, B. Undseth, A. Sammak, G. Scappucci, and L. M. K. Vandersypen, Quantum logic with spin qubits crossing the surface code threshold, *Nature* **601**, 343 (2022).
- [79] R. Blume-Kohout, J. K. Gamble, E. Nielsen, K. Rudinger, J. Mizrahi, K. Fortier, and P. Maunz, Demonstration of qubit operations below a rigorous fault tolerance threshold with gate set tomography, *Nat. Commun.* **8**, 14485 (2017).
- [80] L. Postler, S. Heussen, I. Pogorelov, M. Rispler, T. Feldker, M. Meth, C. D. Marciniak, R. Stricker, M. Ringbauer, R. Blatt, P. Schindler, M. Müller, and T. Monz, Demonstration of fault-tolerant universal quantum gate operations, *Nature* **605**, 675 (2022).
- [81] L. Egan, D. M. Debroy, C. Noel, A. Risinger, D. Zhu, D. Biswas, M. Newman, M. Li, K. R. Brown, M. Cetina, and C. Monroe, Fault-tolerant control of an error-corrected qubit, *Nature* **598**, 281 (2021).
- [82] D. Gottesman, Stabilizer codes and quantum error correction, 1997.
- [83] R. Raussendorf, J. Harrington, and K. Goyal, A fault-tolerant one-way quantum computer, *Ann. Phys. (NY)* **321**, 2242 (2006).
- [84] K. N. Smith, G. Subramanian Ravi, J. M. Baker, and F. T. Chong, in *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)* (IEEE, 2022), p. 1092.
- [85] C. Monroe and J. Kim, Scaling the ion trap quantum processor, *Science* **339**, 1164 (2013).
- [86] M. E. Beverland, P. Murali, M. Troyer, K. M. Svore, T. Hoeffler, V. Kliuchnikov, G. H. Low, M. Soeken, A. Sundaram, and A. Vaschillo, Assessing requirements to scale to practical quantum advantage, *ArXiv:2211.07629*.
- [87] M. D. Hutchings, J. B. Hertzberg, Y. Liu, N. T. Bronn, G. A. Keefe, M. Brink, J. M. Chow, and B. L. T. Plourde, Tunable superconducting qubits with flux-independent coherence, *Phys. Rev. Appl.* **8**, 044003 (2017).
- [88] P. H. Leung and K. R. Brown, Entangling an arbitrary pair of qubits in a long ion crystal, *Phys. Rev. A* **98**, 032318 (2018).
- [89] Q. Liang, M. Kang, M. Li, and Y. Nam, Pulse optimization for high-precision motional-mode characterization in trapped-ion quantum computers, *Quantum Sci. Technol.* **9** (035007).
- [90] M. Fellous-Asiani, J. H. Chai, R. S. Whitney, A. Auffèves, and H. K. Ng, Limitations in quantum computing from resource constraints, *PRX Quantum* **2**, 040335 (2021).
- [91] M. Fellous-Asiani, The resource cost of large scale quantum computing, *ArXiv:2112.04022*.
- [92] <https://www.desmos.com/calculator/jlmygcqrp>.
- [93] S. Bravyi, O. Dial, J. M. Gambetta, D. Gil, and Z. Nazario, The future of quantum computing with superconducting qubits, *J. Appl. Phys.* **132** (2022).
- [94] G. Gentinetta, F. Metz, and G. Carleo, Overhead-constrained circuit knitting for variational quantum dynamics, *Quantum* **8** (2023).
- [95] D. Litinski, Magic state distillation: Not as costly as you think, *Quantum* **3**, 205 (2019).
- [96] J. Gavriel, D. Herr, A. Shaw, M. J. Bremner, A. Paler, and S. J. Devitt, Transversal injection: A method for direct encoding of ancilla states for non-Clifford gates using stabiliser codes, *ArXiv:2211.10046*.
- [97] Y. Akahoshi, K. Maruyama, H. Oshima, S. Sato, and K. Fujii, Partially fault-tolerant quantum computing architecture with error-corrected Clifford gates and space-time efficient analog rotations, *ArXiv:2303.13181*.
- [98] <https://www.desmos.com/calculator/7mbziuf8gd>.
- [99] S. Kimmel, G. H. Low, and T. J. Yoder, Robust calibration of a universal single-qubit gate set via robust phase estimation, *Phys. Rev. A* **92**, 062315 (2015).
- [100] R. Kshirsagar, A. Katabarwa, and P. D. Johnson, On proving the robustness of algorithms for early fault-tolerant quantum computers, *ArXiv:2209.11322*.
- [101] H. Li, H. Ni, and L. Ying, On low-depth quantum algorithms for robust multiple-phase estimation, *ArXiv:2303.08099*.
- [102] Especially in the case where the quantum computation is rate limited by magic state distillation, the computational qubits would be required to idle without accruing error while waiting for T gates or Toffoli gates to be teleported into the computation.
- [103] In the case where the tolerable circuit error rate approaches 1, $p_L \leq (1/G_C) \ln(1/(1-p_C))$ can be used as a tighter bound.
- [104] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, *Contemp. Math.* **305**, 53 (2002).
- [105] A. G. Fowler and C. Gidney, Low overhead quantum computation using lattice surgery, *ArXiv:1808.06709*.
- [106] T. E. O'Brien, M. Streif, N. C. Rubin, R. Santagati, Y. Su, W. J. Huggins, J. J. Goings, N. Moll, E. Kyoseva, M. Degroote, *et al.*, Efficient quantum computation of molecular forces and other energy gradients, *Phys. Rev. Res.* **4**, 043210 (2022).
- [107] Q. Liang, Y. Zhou, A. Dalal, and P. D. Johnson, Modeling the performance of early fault-tolerant quantum algorithms, *ArXiv:2306.17235*.
- [108] H. Ni, H. Li, and L. Ying, On low-depth algorithms for quantum phase estimation, *ArXiv:2302.02454*.
- [109] A. Dutkiewicz, B. M. Terhal, and T. E. O'Brien, Heisenberg-limited quantum phase estimation of multiple eigenvalues with few control qubits, *Quantum* **6**, 830 (2022).
- [110] Z. Ding and L. Lin, Simultaneous estimation of multiple eigenvalues with short-depth quantum circuit on early fault-tolerant quantum computers, *ArXiv:2303.05714*.
- [111] Y. Shen, D. Camps, A. Szasz, S. Darbha, K. Klymko, D. B. Williams-Young, N. M. Tubman, R. Van Beeumen, *et al.*, Estimating eigenenergies from quantum dynamics: A unified noise-resilient measurement-driven approach, *ArXiv:2306.01858*.
- [112] R. Zhang, G. Wang, and P. Johnson, Computing ground state properties with early fault-tolerant quantum computers, *Quantum* **6**, 761 (2022).
- [113] N. H. Stair, R. Huang, and F. A. Evangelista, A multireference quantum Krylov algorithm for strongly

- correlated electrons, *J. Chem. Theory Comput.* **16**, 2236 (2020).
- [114] G. Rendon and P. D. Johnson, Low-depth Gaussian state energy estimation, [ArXiv:2309.16790](https://arxiv.org/abs/2309.16790).
- [115] Z. Ding, C.-F. Chen, L. Lin, *et al.*, Single-ancilla ground state preparation via lindbladians, [ArXiv:2308.15676](https://arxiv.org/abs/2308.15676).
- [116] D. Enshan Koh, G. Wang, P. D. Johnson, and Y. Cao, Foundations for Bayesian inference with engineered likelihood functions for robust amplitude estimation, *J. Math. Phys.* **63**, 052202 (2022).
- [117] Z. Ding, Y. Dong, Y. Tong, and L. Lin, Robust ground-state energy estimation under depolarizing noise, [ArXiv:2307.11257](https://arxiv.org/abs/2307.11257).
- [118] J. Alcazar, A. Cadarso, A. Katabarwa, M. Mauri, B. Peropadre, G. Wang, and Y. Cao, Quantum algorithm for credit valuation adjustments, *New J. Phys.* **24**, 023036 (2022).
- [119] A. Katabarwa, A. Kunitsa, B. Peropadre, and P. Johnson, Reducing runtime and error in VQE using deeper and noisier quantum circuits, [ArXiv:2110.10664](https://arxiv.org/abs/2110.10664).
- [120] T. Giurgica-Tiron, S. Johri, I. Kerenidis, J. Nguyen, N. Pisenti, A. Prakash, K. Sosnova, K. Wright, and W. Zeng, Low-depth amplitude estimation on a trapped-ion quantum computer, *Phys. Rev. Res.* **4**, 033034 (2022).
- [121] Note that the so-called semiclassical Fourier transform can be used to reduce the ancilla count to one, though it requires midcircuit measurement, reset, and feed-forward at the logical level.
- [122] R. D. Somma, Quantum eigenvalue estimation via time series analysis, *New J. Phys.* **21**, 123025 (2019).
- [123] A. M. Childs and N. Wiebe, Hamiltonian simulation using linear combinations of unitary operations, [ArXiv:1202.5822](https://arxiv.org/abs/1202.5822).
- [124] E. Campbell, Random compiler for fast Hamiltonian simulation, *Phys. Rev. Lett.* **123**, 070503 (2019).
- [125] M. Steudtner, S. Morley-Short, W. Pol, S. Sim, C. L. Cortes, M. Loipersberger, R. M. Parrish, M. Degroote, N. Moll, R. Santagati, *et al.*, Fault-tolerant quantum computation of molecular observables, [ArXiv:2303.14118](https://arxiv.org/abs/2303.14118).
- [126] E. Knill, G. Ortiz, and R. D. Somma, Optimal quantum measurements of expectation values of observables, *Phys. Rev. A* **75**, 012328 (2007).
- [127] N. H. Stair, C. L. Cortes, R. M. Parrish, J. Cohn, and M. Motta, Stochastic quantum Krylov protocol with double-factorized Hamiltonians, *Phys. Rev. A* **107**, 032414 (2023).
- [128] S. Pathak, A. E. Russo, S. K. Seritan, and A. D. Baczewski, Quantifying T -gate-count improvements for ground-state-energy estimation with near-optimal state preparation, *Phys. Rev. A* **107**, L040601 (2023).
- [129] K. Gratsea, C. Sun, and P. D. Johnson, When to reject a ground state preparation algorithm, [ArXiv:2212.09492](https://arxiv.org/abs/2212.09492).
- [130] K. Gratsea, J. S. Kottmann, P. D. Johnson, and A. A. Kunitsa, Comparing classical and quantum ground state preparation heuristics, [ArXiv:2401.05306](https://arxiv.org/abs/2401.05306).
- [131] M. A. Nielsen and I. L. Chuang, Quantum computation and quantum information: 10th anniversary edition (2011).
- [132] <https://www.desmos.com/calculator/nf43nafwet>.
- [133] <https://www.desmos.com/calculator/7fwvz6fxgb>.
- [134] Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O'Brien, Quantum error mitigation, [ArXiv:2210.00921](https://arxiv.org/abs/2210.00921).
- [135] X. Xiao, J. K. Freericks, and A. F. Kemper, Robust measurement of wave function topology on NISQ quantum computers, *Quantum* **7**, 987 (2023).
- [136] T. E. Brien, B. Tarasinski, and B. M. Terhal, Quantum phase estimation of multiple eigenvalues for small-scale (noisy) experiments, *New J. Phys.* **21**, 023022 (2019).
- [137] P. D. Johnson, A. A. Kunitsa, J. F. Gonthier, M. D. Radin, C. Buda, E. J. Doskocil, C. M. Abuan, and J. Romero, Reducing the cost of energy estimation in the variational quantum eigensolver algorithm with robust amplitude estimation, [ArXiv:2203.07275](https://arxiv.org/abs/2203.07275).
- [138] I. H. Kim, Y.-H. Liu, S. Pallister, W. Pol, S. Roberts, and E. Lee, Fault-tolerant resource estimate for quantum chemical simulations: Case study on Lithium battery electrolyte molecules, *Phys. Rev. Res.* **4** (2022).
- [139] C. Piveteau, D. Sutter, S. Bravyi, J. M. Gambetta, and K. Temme, Error mitigation for universal gates on encoded qubits, *Phys. Rev. Lett.* **127**, 200505 (2021).
- [140] Y. Suzuki, S. Endo, K. Fujii, and Y. Tokunaga, Quantum error mitigation as a universal error reduction technique: Applications from the NISQ to the fault-tolerant quantum computing eras, *PRX Quantum* **3**, 010345 (2022).
- [141] <https://www.desmos.com/calculator/cnh0vchq6l>.