Fault-Tolerant Execution of Error-Corrected Quantum Algorithms

Michael A. Perlin,[1,*] Zichang He,[1,†] Anthony Alexiades Armenakas,[1] Pablo Andres-Martinez,[2] Tianyi Hao,[1] Dylan A. Herman,[1] Yuwei Jin,[1] Karl Mayer,[3] Chris N. Self,[2] David Amaro,[2] Ciaran Ryan-Anderson,[3] and Ruslan Shaydulin[1,‡]

[1]*Global Technology Applied Research, JPMorganChase, New York, New York 10017, USA*
[2]*Quantinuum, Partnership House, Carlisle Place, London SW1P 1BX, United Kingdom*
[3]*Quantinuum, 303 South Technology Court, Broomfield, Colorado 80021, USA*

(Received 6 April 2026; revised 23 August 2026; accepted 9 October 2026; published 19 November 2026)

Scaling up quantum algorithms to tackle high-impact problems in science and industry requires quantum error correction and fault tolerance. While progress has been made in experimentally realizing error-corrected primitives, the end-to-end execution of logical quantum algorithms using only fault-tolerant (FT) components has remained out of reach. We demonstrate the FT and error-corrected execution of two quantum algorithms, the quantum approximate optimization algorithm (QAOA) and the Harrow-Hassidim-Lloyd algorithm applied to the Poisson equation, on Quantinuum H2 and Helios trapped-ion quantum processors using the $[\![7,1,3]\!]$ Steane code. For QAOA circuits on five and six logical qubits, we show performance improvements from increasing the number of QAOA layers and the number of T gates used to approximate logical rotations, despite increased physical circuit complexity. We further show that QAOA circuits with up to eight logical qubits and nine logical T gates perform similarly to unencoded circuits. For the largest QAOA circuits we run, with 12 logical (97 physical) qubits and 2132 physical two-qubit gates, we still observe better-than-random performance. Finally, we show that adding active quantum error correction cycles and increasing the repeat-until-success limit of state-preparation subroutines can improve the performance of a quantum algorithm, thereby demonstrating critical capabilities of scalable FT quantum computation. Our results are enabled by a FT logical T-gate implementation with an infidelity of $2.7(4) \times 10^{-3}$ and dynamic circuits with measurement-dependent feedback. Our work demonstrates near-breakeven performance of complex, error-corrected algorithmic quantum circuits using only FT components.

DOI: 10.1103/gnq3-c2df

Subject Areas: Quantum Information

I. INTRODUCTION

Realizing a broad class of high-impact applications of quantum computers at scale is widely anticipated to require quantum error correction (QEC) and fault tolerance. Recent progress in quantum hardware has enabled demonstrations of below-threshold quantum memory [1,2], with the logical error rate decreasing with code distance. Furthermore, quantum primitives have been demonstrated below breakeven [3,4], with the logical error rate falling below the physical error rate for small error-correcting codes of fixed distance.

A central outstanding challenge is demonstrating fault-tolerant execution of complex algorithmic circuits below breakeven, with the encoded circuit outperforming the same circuit executed directly using physical operations. Realizing such a demonstration requires all components of the QEC stack to work together to support application execution. While the success of recent component-level demonstrations suggests this milestone may be achievable on near-term devices, many obstacles remain that prevent straightforward combination of components in an end-to-end pipeline.

One obstacle is the number of different primitives required to execute quantum algorithms in their entirety and the variety of options available for implementing these primitives. This challenge is particularly acute in architectures that support qubit movement, such as trapped ions [5–7] and neutral atoms [8–11], as the flexibility afforded by qubit movement allows for a broad selection of quantum error-correcting codes [12–14], gate implementations [15], and approaches to fault tolerance [16–19].

We focus on the Steane code [20], which is one of the smallest and simplest error-correcting codes. The

*Contact author: michael.perlin@jpmchase.com
†Contact author: zichang.he@jpmchase.com
‡Contact author: ruslan.shaydulin@jpmchase.com

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

development of specialized circuits for performing QEC [21] and optimized gadgets for universal computation [22] with the Steane code enabled early demonstrations of real-time fault-tolerant QEC [23] and the realization of a universal fault-tolerant gate set [24]. Further hardware improvements led to the achievement of breakeven (that is, better-logical-than-physical) performance for select Clifford [4] and non-Clifford [25] gates. At the same time, the Steane code has enabled *application-level* error-corrected hardware benchmarking, notably including the three-qubit quantum Fourier transform [26] and a quantum computation of molecular energies [27], mirroring the standard practice of application-level benchmarking of classical computing hardware [28]. Because of the overheads of fully fault-tolerant execution, however, all demonstrations of logically encoded quantum algorithms (including Refs. [26,27]) have, to date, been performed only at the component level [8,11,24,29–32], using non-fault-tolerant components [26,27], without error correction [9,33,34], or with a nonuniversal gate set [3].

In this work, we demonstrate and benchmark the execution of error-corrected quantum algorithms using only fault-tolerant gadgets on trapped-ion quantum computers based on the quantum charge-coupled device (QCCD) [5] architecture, namely, H2-1, H2-2 [6], and Helios [7]. Our emphasis is on how these gadgets integrate into complete algorithms rather than on the performance of any single gadget in isolation. To this end, we first review and benchmark, at the component level, circuit gadgets that enable universal quantum computation with the Steane code. Among the components that enable this demonstration, we introduce a fault-tolerant logical T gate that outperforms the prior state of the art on hardware, achieving a fidelity of 0.960(6) for a Ramsey sequence with $16T$ gates on H2-1, corresponding to an infidelity of approximately $2.7(4) \times 10^{-3}$ per T gate. For comparison, prior work (Fig. 5, T method 2 in Ref. [26]) demonstrated a non-fault-tolerant logical T gate with infidelity 14×10^{-3}. We additionally benchmark several methods for performing a QEC cycle and find that a Steane-type gadget that teleports data onto a fresh code block outperforms a previously used flag-fault-tolerant protocol.

Putting these ingredients together, we execute the quantum approximate optimization algorithm (QAOA) [35,36] for small instances of the low autocorrelation binary sequences (LABS) problem [37,38] and a portfolio optimization problem instance, as well as the Harrow-Hassidim-Lloyd (HHL) algorithm [39] for a structured toy problem, namely, the Poisson equation on a lattice. Notably, we observe improved QAOA performance with increasing QAOA depth and when compiling to more T gates, the noisiest element of our logical instruction set. Encoded circuits with few T gates consistently perform comparably to their unencoded versions. This near-breakeven performance is made possible by both hardware improvements and the use of optimized fault-tolerant gadgets.

Moreover, we find that increasing the number of allowed attempts in repeat-until-success (RUS) subroutines has little effect on the performance of QAOA on H2-2 and no effect on Helios while reducing the postselection discard rate to nearly zero, thereby demonstrating an essential capability for scalable fault-tolerant quantum computation. Our HHL benchmarks similarly demonstrate the suppression of state-preparation errors with RUS and the improvement of logical circuit fidelity with active QEC cycles. Through careful examination of HHL data, we identify dynamic program compilation artifacts as performance bottlenecks for the algorithm—highlighting the importance of application-level benchmarking to reveal system-level behaviors that may not manifest in component benchmarks.

The remainder of this paper is structured as follows. We review the Steane code and circuit gadgets necessary for universal fault-tolerant quantum computation in Sec. II, where we additionally introduce improved methods for applying a fault-tolerant T gate. We review the QAOA algorithm, the LABS problem, and binary portfolio optimization as well as the HHL algorithm and a toy problem suitable for near-term implementation in Sec. III. We benchmark several Steane gadgets in isolation both in emulators and on hardware in Sec. IV A and perform our application-level benchmarking in Sec. IV B. Finally, we close with a discussion in Sec. V.

II. FAULT-TOLERANT QUANTUM COMPUTING WITH THE STEANE CODE

We begin by reviewing the ingredients necessary for universal fault-tolerant quantum computation with the Steane code and the various options for implementing them on trapped-ion quantum processors with qubit movement. For the purposes of this work, we say that a circuit gadget or protocol is fault tolerant if every midcircuit error propagates to a correctable error at the end of the protocol. Here, midcircuit errors include single-qubit errors after single-qubit gates, two-qubit errors after two-qubit gates, and single-qubit memory errors. For simplicity, we do not explicitly consider the correction of leakage errors, though we note that (unheralded) leakage errors are equivalent to Pauli errors when teleporting quantum data onto fresh qubits. This operational definition of fault-tolerant gadgets is sufficient for the Steane code, which makes no promise of correcting more than one error between any two rounds of QEC.

We remark that the above notion of fault tolerance is a property of an individual gadget. Guaranteeing fault tolerance at the level of a full computation requires additional measures, such as interleaving QEC cycles to form extended rectangles [40] or correlated decoding [16,17]. The algorithms we run are built entirely from fault-tolerant gadgets, but we do not make them fault tolerant in this

stronger, whole-computation sense; we return to this point, and its consequences for our benchmarks, in Appendix A. We refer the reader to Refs. [40,41] for more general and nuanced definitions and discussions of fault tolerance.

Throughout this work, we denote a single-qubit identity operator by $I = |0\rangle\langle 0| + |1\rangle\langle 1|$, a Pauli-$X$ operator by $X = |0\rangle\langle 1| + |1\rangle\langle 0|$, and a Pauli-$Z$ operator by $Z = |0\rangle\langle 0| - |1\rangle\langle 1|$ and let $Y = iXZ$. We denote the action of a single-qubit operator O on physical qubit j by O_j. For any n-bit string $\boldsymbol{b} = (b_1, b_2, \ldots, b_n) \in \{0,1\}^n$, we let $O(\boldsymbol{b}) = \prod_{j:b_j=1} O_j$ denote the action of O on the qubits indicated by $\boldsymbol{b}$. To disambiguate operators that act on physical and logical degrees of freedom, in this section we use an overline to indicate that operators act on encoded logical qubits, as in $\bar{O}$, though we drop this notation in Sec. IV when there is no ambiguity. Finally, we refer to the physical qubits that encode a logical qubit as the *data qubits* of an error-correcting code.

A. The Steane code

The Steane code [20] is the smallest Calderbank-Shor-Steane quantum error-correcting code [20,42] (as well as the smallest error-correcting topological and color code [43,44]) and has code parameters $[\![7,1,3]\!]$, meaning that it encodes one logical qubit into seven physical qubits and has code distance 3. The Steane code can correct any single-qubit Pauli error as well as two-qubit errors of mixed X/Z type, such as X_0Z_4. Logical states of the Steane code are $+1$ eigenvectors of all stabilizers $s \in \mathcal{S}$, where the stabilizer group $\mathcal{S}$ consists of all products of stabilizer generators $g \in G = G_X \cup G_Z$, and the generator subgroup $G_P = \{P(\boldsymbol{b}) : \boldsymbol{b} \in \mathrm{rows}(h)\}$ is defined in terms of the rows of the parity check matrix of the classical [7, 4, 3] Hamming code,

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

The logical X and Z operators of the Steane code are

$$\bar{X} = \prod_{j=0}^{6} X_j \simeq X_0X_1X_2, \qquad \bar{Z} = \prod_{j=0}^{6} Z_j \simeq Z_0Z_1Z_2, \tag{2}$$

where $\simeq$ denotes equality up to multiplication by stabilizers, and throughout this work we index the qubits of a Steane code block from 0 to 6. Figure 1 shows a geometric representation of the Steane code.

We note that there are different conventions in the literature regarding the ordering of qubits for the Steane code. We use the ordering in (for example) Refs. [20,22,45–47], which differs from that in Refs. [23,26,27,48]. The map from qubit index in Refs. [23,26,27,48] to that here and in Refs. [20,22,45–47] is $\{4 \to 0, 5 \to 1, 6 \to 2, 3 \to 3, 2 \to 4, 1 \to 5, 0 \to 6\}$.

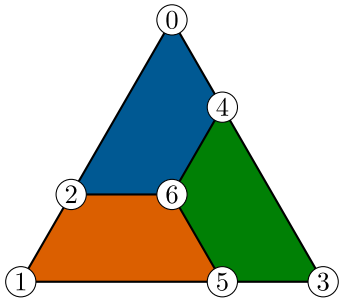


FIG. 1. A geometric representation of the Steane code. Nodes represent qubits. Colored cells identify stabilizer group generators, which are obtained by taking the product of either Pauli-X or Pauli-Z operators on the adjacent nodes, such as $X_1X_2X_6X_5$. The product of either Pauli-X or Pauli-Z operators on any exterior edge of the triangle yields, respectively, a logical Pauli-X or Pauli-Z operator, such as $Z_0Z_4Z_3$.

B. Clifford gates and initialization

The Steane code has a transversal logical Hadamard gate $\bar{H} = \prod_{j=0}^{6} H_j$ and a transversal logical phase gate $\bar{S} = \prod_{j=0}^{6} S_j^\dagger$, where $H = |+\rangle\langle 0| + |-\rangle\langle 1|$ and $S = |0\rangle\langle 0| + i|1\rangle\langle 1|$. These gates suffice to make all single-qubit Clifford gates transversal in the Steane code. A logical $\overline{\mathrm{CNOT}}$ gate of the Steane code is implemented by broadcasting physical $\mathrm{CNOT} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X$ gates pairwise across the data qubits of two Steane code blocks. That is, if logical qubits $\bar{0}$ and $\bar{1}$ are supported, respectively, on physical qubits $0, 1, \ldots, 6$ and $7, 8, \ldots, 13$, then $\overline{\mathrm{CNOT}}_{\bar{0},\bar{1}} = \prod_{j=0}^{6} \mathrm{CNOT}_{j,j+7}$. The fact that these logical gates are transversal, meaning that every logical gate acts on at most one data qubit of a Steane code block, makes them automatically fault tolerant, because there is no mechanism for a gate error on one data qubit in a Steane code block to propagate to another qubit within the same code block. Transversality of the logical H and CNOT implies that the logical controlled-Z gate, $\mathrm{CZ} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Z$, is also transversal.

To initialize logical states of the Steane code, we use the repeat-until-success protocol for $|\bar{0}\rangle$-state preparation in Ref. [22], shown in Fig. 2, which uses an ancilla qubit to measure $\bar{Z}$ and discard faulty outcomes. This circuit has the same two-qubit gate count as, for example, fault-tolerant circuits discovered with reinforcement learning [49]. Other Pauli eigenstates of the Steane code, such as $|\overline{+}\rangle$, can be prepared by initializing $|\bar{0}\rangle$ and applying a logical Clifford gate.

C. Measurement and decoding

A logical Z-basis measurement of a Steane code block is performed by measuring all data qubits in the Z basis,

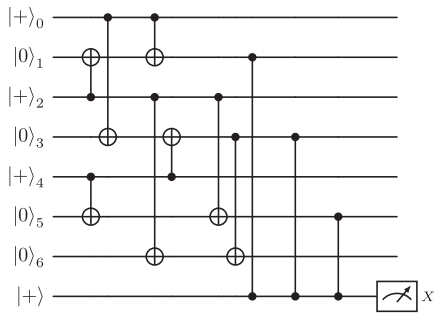
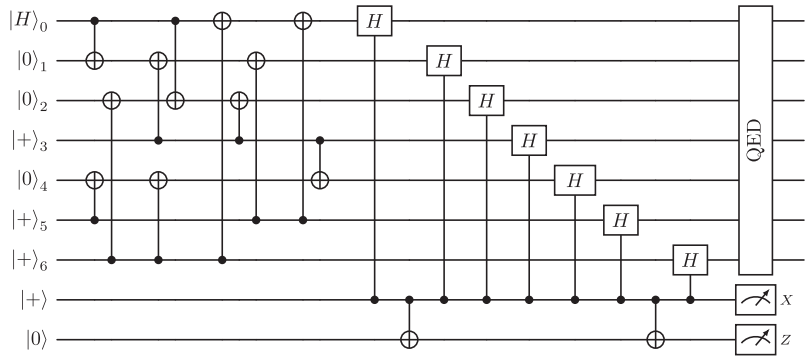
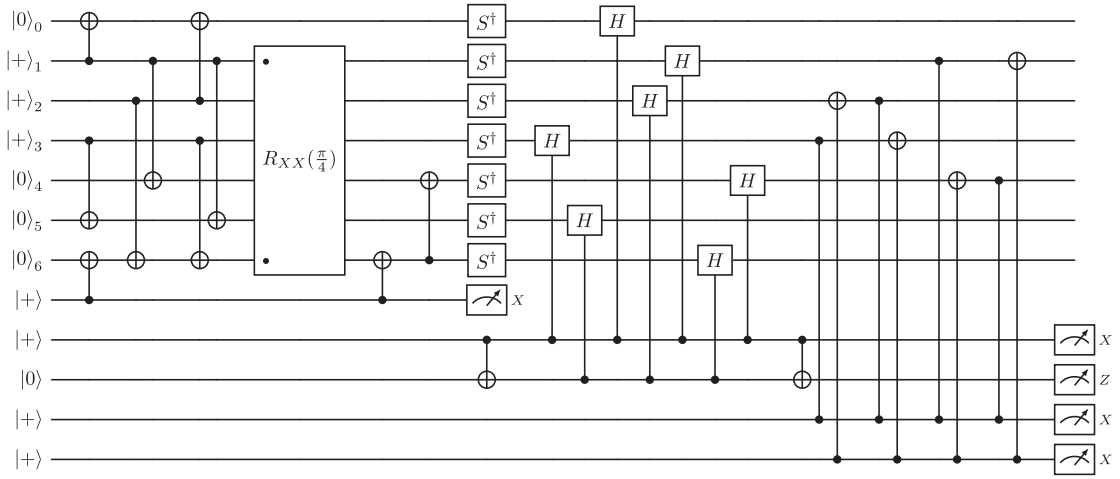


FIG. 2. Repeat-until-success protocol to fault-tolerantly prepare a logical $|\bar{0}\rangle$ state of the Steane code using CNOT and CZ gates. The bottom qubit is an ancilla ("flag") qubit that is physically measured in the X basis to read out the value of the logical $\bar{Z} \simeq Z_1 Z_3 Z_5$ operator. This protocol "succeeds" if the flag measurement outcome is 0 (that is, if the flag qubit is found to be in $|+\rangle$ upon measurement in the X basis); otherwise, the qubits in this circuit are reset and the protocol is repeated until success.

which yields a seven-bit string $\boldsymbol{b}_{\text{measured}}$. This bit string may generally contain errors, which can be diagnosed by the *error syndrome* $\boldsymbol{s} = h \cdot \boldsymbol{b}_{\text{measured}}$ (mod 2). In turn, the syndrome $\boldsymbol{s}$ needs to be *decoded* to infer a correction $\boldsymbol{b}_{\text{measured}} \to \boldsymbol{b}_{\text{corrected}}$. In the case of the Steane code, a nonzero syndrome $\boldsymbol{s}$ corresponds to a bit-flip error on bit $\texttt{int}(\boldsymbol{s}) - 1$ of $\boldsymbol{b}_{\text{measured}}$ [that is, the measurement bit associated with qubit $\texttt{int}(\boldsymbol{s}) - 1$], where $\texttt{int}(\boldsymbol{s})$ is the integer encoded by the bit string $\boldsymbol{s}$, with (for example) $\texttt{int}[(0,0,1)] = 1$ and $\texttt{int}[(0,1,0)] = 2$. This simple decoding procedure is a consequence of the facts that (a) only single-qubit errors are correctable in the Steane code, and (b) the nonzero bit string $\boldsymbol{v}$ is the $[\texttt{int}(\boldsymbol{v}) - 1]$th column of h in Eq. (1). After correcting the error inferred from $\boldsymbol{s}$ by flipping the corresponding bit in $\boldsymbol{b}_{\text{measured}}$ to obtain $\boldsymbol{b}_{\text{corrected}}$, the parity of $\boldsymbol{b}_{\text{corrected}}$ is the logical Z-basis measurement outcome. A logical X-basis measurement can



(a) Logical $|\overline{H}\rangle$ state preparation by directly encoding a physical $|H\rangle$ state.



(b) Specialized $|\overline{H}\rangle$-state preparation circuit.

FIG. 3. Repeat-until-success protocols to fault-tolerantly prepare $|\bar{H}\rangle$. Measurements are performed in the Z or X basis as indicated. (a) A circuit that non-fault-tolerantly encodes a single-qubit $|H\rangle$ state into $|\bar{H}\rangle$, fault-tolerantly measures $\bar{H}$, and runs a full round of QED that consists of six stabilizer measurements. (b) A specialized circuit to fault-tolerantly prepare $|\bar{H}\rangle$ using a shorter non-fault-tolerant $|\bar{H}\rangle$ preparation circuit, two qubits to flag midcircuit errors, a fault-tolerant measurement of $\bar{H}$, and two stabilizer measurements. Here, $R_{XX}(\theta) = e^{-i\theta X \otimes X/2}$ is a two-qubit Pauli rotation on the qubits indicated by dots (•). The circuit here has been expanded for clarity but can be implemented with a two-qubit gate depth of 9.

be performed identically by measuring all data qubits in the X basis and using the same decoding procedure.

D. Non-Clifford gates

Universal quantum computation requires non-Clifford gates. The presence of a transversal logical Hadamard gate $\bar{H}$ enables a simple strategy to fault-tolerantly prepare a non-Clifford state: the $+1$ eigenstate of $\bar{H}$, which we denote $|\bar{H}\rangle$. The basic idea is to first prepare $|\bar{H}\rangle$ non-fault-tolerantly and then perform a fault-tolerant measurement of $\bar{H}$ and some quantum error detection (QED) to project out preparation errors. This is a repeat-until-success protocol: nonzero measurement outcomes trigger a reset and restart of the protocol. Figure 3(a) shows a straightforward implementation of this protocol, which has featured in prior works [22,24,26].

We construct an improved circuit to fault-tolerantly prepare the $|\bar{H}\rangle$ state in Fig. 3(b). This circuit uses the non-fault-tolerant $|\bar{T}\rangle$-state-preparation circuit in Eq. (B13) of Ref. [27], which (similarly to the circuit in Fig. 2 of Ref. [47]) requires only nine two-qubit gates, as opposed to the 11 two-qubit gates required for the encoding circuit in Fig. 3(a). Similar circuits can be constructed by propagating representations of the $\bar{X}$ operator (that may differ by stabilizer factors) backward through a $|\bar{0}\rangle$-state-preparation circuit to find locations at which the backpropagated $\bar{X}$ is a Pauli string with weight 2. Inserting a pair-Pauli rotation gate at such a location makes the circuit prepare the state $R_{\bar{X}}(\theta)|\bar{0}\rangle = \exp[-i(\theta/2)\bar{X}]|\bar{0}\rangle = \cos(\theta/2)|\bar{0}\rangle + \sin(\theta/2)|\bar{1}\rangle$. In addition, the specialized circuit in Fig. 3(b) uses one qubit to flag midcircuit errors and measures only two stabilizers for QED after the measurement of $\bar{H}$. We provide a detailed discussion of Fig. 3(b) and its construction in Appendix B.

We note that Fig. 6 in Ref. [45] and Fig. 11 in Ref. [46] provide deterministic protocols to fault-tolerantly prepare a $|\bar{H}\rangle$ state of the Steane code. These strategies use additional flag qubits to diagnose midcircuit errors. We focus on repeat-until-success strategies in this work and refer to the maximum number of attempts allowed in a repeat-until-success subroutine before discarding the entire circuit and restarting a computation as the "RUS limit."

After preparing $|\bar{H}\rangle$, this state can be rotated into the more commonly used $|\bar{T}\rangle = \bar{T}|\overline{+}\rangle$ state as $|\bar{T}\rangle = \overline{HS}|\bar{H}\rangle$. Here, $T = |0\rangle\langle 0| + e^{i\pi/4}|1\rangle\langle 1|$. The $|\bar{T}\rangle$ state can, in turn, be consumed by one of the gadgets in Fig. 4(a) or 4(b) to apply a $\bar{T}$ gate and thereby complete a Clifford + T gate set for the Steane code.

E. Quantum error correction

Finally, we discuss strategies for performing QEC and QED in the Steane code. In principle, to perform QEC one needs to measure a complete set of stabilizers that generate the code's stabilizer group. The syndrome (bit string) obtained from these stabilizer measurements can then be decoded to infer a data qubit error. In principle, each stabilizer can be measured using a gadget similar to that in Fig. 5. However, some care is needed to make sure that stabilizer measurement gadgets are themselves fault tolerant. Moreover, redundant syndrome measurements are required to diagnose measurement errors.

The paradigm of *flag fault tolerance* [50,51] provides a general scheme for fault-tolerant stabilizer measurements. This paradigm typically makes careful use of additional flag qubits whose measurement outcomes provide sufficient information to diagnose midcircuit errors. In the case of the Steane code, Ref. [21] developed a specialized flag-fault-tolerant syndrome measurement protocol that uses some syndrome measurement qubits as flags for others, altogether requiring only six ancilla qubits to measure six stabilizers (and requiring only three ancilla qubits at a

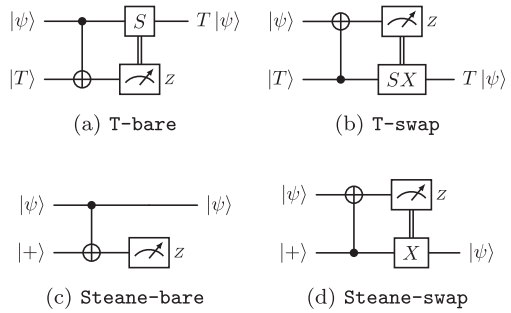


FIG. 4. (a),(b) Two gadgets that consume a $|T\rangle = T|+\rangle$ state to apply a T gate to an arbitrary single-qubit state $|\psi\rangle$. At the level of circuit identities, these gadgets can be generalized by replacing $T \to R_Z(\theta)$ and $S \to R_Z(2\theta)$ for any angle θ, but the resulting circuits may not have fault-tolerant implementations. (c),(d) Analogous gadgets (with $\theta = 0$) that implement a fault-tolerant error detection cycle for bit-flip (X) errors. For (c), detected errors can be either actively corrected or tracked in software. For (d), the conditional logical X correction can similarly be tracked in software by updating a Pauli frame. Gadgets (b) and (d) have the benefit of mitigating leakage errors in $|\psi\rangle$ and correcting physical X errors passively, as these errors do not propagate to the output state of the gadget. Changing bases as $Z \leftrightarrow X$, $|+\rangle \leftrightarrow |0\rangle$, and flipping the orientation of the CNOT yields gadgets for fault-tolerantly implementing a $T_X = HTH$ gate and detecting phase-flip errors.

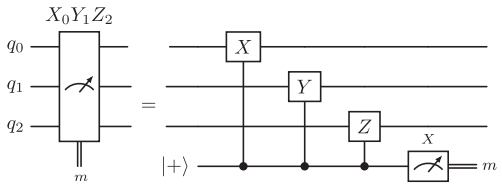


FIG. 5. Non-fault-tolerant gadget to measure the Pauli string $X_0Y_1Z_2$, writing the result to the bit m. Single-qubit errors in the middle of the decomposition on the right can propagate to higher-weight errors on qubits q_0, q_1, and q_2.

time). If an initial round of flagged stabilizer measurements yields a nontrivial syndrome, this protocol triggers a second round of stabilizer measurements, and the combined syndromes from the two measurement rounds suffice to diagnose and correct individual midcircuit errors in a fault-tolerant manner. This protocol was experimentally implemented in Ref. [23]. We review and provide additional details about the flag-fault-tolerant QEC protocol used in this work, including explicit circuits, in Appendix C.

An alternative general strategy for fault-tolerant QEC is Steane error correction (note the naming collision: Steane error correction need not be performed with the Steane code). Steane QEC can be thought of as "copying" the physical errors in one code block onto an ancilla code block —crucially, without affecting the logical state of either code block—and then measuring out the ancilla code block to diagnose the errors. This strategy is shown in Fig. 4(c) and can be recovered from a generalization of the T-gate teleportation gadget in Fig. 4(a). Physical X errors on the top (logical) qubit propagate through the transversal CNOT to physical X errors on the bottom (logical) qubit, which, in turn, flip Z-basis measurements of the ancilla, thereby revealing the location of the X errors. However, the CNOT does not affect the logical states of these qubits: $\mathrm{CNOT}(|\psi\rangle \otimes |+\rangle) = |\psi\rangle \otimes |+\rangle$. Preparing the ancilla qubit in $|\bar{0}\rangle$, flipping the orientation of the CNOT, and measuring the ancilla qubit in the X basis similarly diagnoses physical phase-flip (Z) errors. Diagnosed errors can be either actively corrected or virtually tracked in software. We refer to the execution of Fig. 4(c) followed by its Z-correction variant as the Steane-bare strategy for QEC in this work.

A teleportationlike variant of Steane QEC, shown in Fig. 4(d), can similarly be recovered from a generalization of the T-gate teleportation gadget in Fig. 4(b). This strategy corrects errors *passively*: physical X errors on the top (logical) qubit do not propagate to the bottom (logical) qubit and, instead, are "decoded away" in the logical Z-basis measurement of the top qubit. As with ordinary Steane QEC, appropriately swapping X and Z bases in Fig. 4(d) yields a gadget for correcting phase-flip (Z) errors. We refer to the execution of Fig. 4(d) followed by its Z-correction variant as the Steane-swap strategy for QEC in this work.

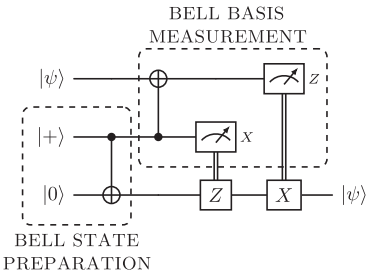


FIG. 6. Standard teleportation circuit for Knill QEC.

Finally, QEC can also be performed with full quantum state teleportation, which is commonly referred to as Knill QEC. A standard circuit for this strategy is shown in Fig. 6. A minor modification of this strategy, in which the second logical CNOT is performed before the first, reduces Knill QEC to a parallelized version of (two complementary rounds of) Steane-swap QEC.

III. APPLICATION BENCHMARKS

Here, we describe the algorithms that we use for application-level benchmarking of the Steane code on the H2-1, H2-2 [6], and Helios [7] trapped-ion quantum computers.

A. QAOA

We first consider the QAOA [35,36], which takes as input (a) a *cost Hamiltonian* H_C that encodes an N-bit optimization problem and (b) two real-valued vectors $\boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{R}^p$, and prepares the state $|\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle = U_{\mathrm{QAOA}}(\boldsymbol{\beta}, \boldsymbol{\gamma})|+\rangle^{\otimes N}$, where p is referred to as the *QAOA depth* and the unitary

$$U_{\mathrm{QAOA}}(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \prod_{j=1}^{p} e^{-i\beta_j H_M} e^{-i\gamma_j H_C} \tag{3}$$

additionally depends on a choice of *mixing Hamiltonian* H_M that is typically taken to be a transverse field, $H_M = \frac{1}{2}\sum_j X_j$. The output of QAOA is a list of bit strings obtained by repeatedly preparing and measuring the state $|\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle$ in the computational basis. The *success probability* of a QAOA state $|\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle$ is the probability of measuring the optimal solution to the optimization problem or the ground state of H_C.

The cost Hamiltonian H_C for a cost function $C: \{0,1\}^N \to \mathbb{R}$ is typically constructed by energetically penalizing the bit strings according to their cost, for example, with $H_C = \sum_{\boldsymbol{x} \in \{0,1\}^N} C(\boldsymbol{x})|\boldsymbol{x}\rangle\langle\boldsymbol{x}|$. The QAOA circuit is equivalent, up to a choice of $\boldsymbol{\beta}, \boldsymbol{\gamma}$, to the Trotterized evolution of an Ising model with a transverse field.

Extensive theoretical and numerical evidence suggests that QAOA can provide a polynomial [52–55] and, in some restricted cases, exponential [56,57] speedup over state-of-the-art classical algorithms. Though QAOA can be viewed as a variational algorithm in which the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are optimized inside a classical control loop, for some problem classes it is possible to obtain fixed (instance-independent) parameters [52,53,56,58–62] that nonetheless yield an advantage over classical state-of-the-art methods, as is the case for the problems considered in this work [53,58]. Consequently, in this work we compile QAOA with fixed parameters $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$.

1. LABS

The first problem that we tackle with QAOA is the LABS problem [37,38], which has applications in communications engineering [38,63] and for which there exists evidence of a scaling advantage for QAOA over state-of-the-art classical methods [53]. The objective of LABS is to find a binary sequence $\boldsymbol{s} = (s_1, s_2, \ldots, s_N) \in \{+1, -1\}^N$ that minimizes the autocorrelation energy

$$E_{\mathrm{LABS}}(\boldsymbol{s}) = \sum_{k=1}^{N-1} A_k(\boldsymbol{s})^2, \qquad A_k(\boldsymbol{s}) = \sum_{j=1}^{N-k} s_j s_{j+k}. \quad (4)$$

The QAOA cost Hamiltonian H_{LABS} for LABS is obtained by promoting each variable s_j in Eq. (4) to a single-qubit Pauli-Z operator Z_j. The LABS problem becomes classically intractable to solve exactly at approximately 100 binary variables, making it a promising candidate for demonstrating quantum advantage in optimization [64–67]. Reference [53] provides optimized QAOA parameters for LABS for small N, and instance-independent parameters for large N, which we use in this work.

2. Portfolio optimization

The second problem we tackle with QAOA is portfolio optimization. We consider the simplest version of this problem, namely, binary unconstrained mean-variance portfolio optimization, which takes the form of a quadratic unconstrained binary optimization problem. The goal is to find a binary vector $\boldsymbol{s} = (s_1, s_2, \ldots, s_N) \in \{0, 1\}^N$ that indicates whether a given asset is included or excluded from a portfolio to minimize the objective function

$$C(\boldsymbol{s}) = q \sum_{ij} W_{ij} s_i s_j - \sum_i \mu_i s_i, \quad (5)$$

where $\boldsymbol{\mu} \in \mathbb{R}^N$ denotes the vector of expected returns for the assets, $W \in \mathbb{R}^{N \times N}$ is the covariance matrix between the N assets, and $q > 0$ is a risk factor that balances risk and return in the objective function. The corresponding QAOA cost Hamiltonian H_C is obtained by promoting each variable s_j in Eq. (5) to the single-qubit projector $|1\rangle\langle 1|_j = \frac{1}{2}(1 - Z_j)$. Letting $C_{\min} = \min_{\boldsymbol{s}} C(\boldsymbol{s})$ and $C_{\max} = \max_{\boldsymbol{s}} C(\boldsymbol{s})$ denote the extrema of C, the approximation ratio achieved by a QAOA state $|\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle$ is

$$\mathrm{AR}(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \frac{\langle \boldsymbol{\beta}, \boldsymbol{\gamma} | H_C | \boldsymbol{\beta}, \boldsymbol{\gamma} \rangle - C_{\min}}{C_{\max} - C_{\min}}. \quad (6)$$

Portfolio optimization problem instances are defined by a choice of expected returns $\boldsymbol{\mu}$ and covariance matrix W for a basket of assets, which we generate from historical equity price data using QOKIT [68].

B. HHL

The HHL algorithm [39] was proposed for solving a quantum analog of a traditional linear system of equations and can be stated as follows. Given an error tolerance $\epsilon > 0$ and appropriate access to a vector $\boldsymbol{b} \in \mathbb{C}^M$ and matrix $A \in \mathbb{C}^{M \times N}$ for which $M \leq N$ and the operator norm $\|A\| \leq 1$, prepare a quantum state $|\boldsymbol{y}\rangle \in \mathbb{C}^N$ that is ϵ-close to a solution $\boldsymbol{x}_\star$ of $A\boldsymbol{x} = \boldsymbol{b}$, meaning $\||\boldsymbol{y}\rangle - |\boldsymbol{x}_\star\rangle\| < \epsilon$ for the normalized state $|\boldsymbol{x}_\star\rangle$ that is proportional to $\boldsymbol{x}_\star$ (where $\||\psi\rangle\| = \sqrt{\langle\psi|\psi\rangle}$). This task is commonly referred to as the quantum linear systems problem (QLSP).

Various algorithmic improvements have extended the original approach of HHL [69–71], eventually achieving a run-time that is asymptotically optimal in the number of queries to A and $\boldsymbol{b}$, namely, $\Theta[\kappa \log(1/\epsilon)]$, where $1/\kappa$ is the smallest nonzero singular value of A [72]. Altogether, algorithms for the QLSP with sparse matrices (that is, matrices whose maximum number of nonzero entries per row and column is independent of M and N) have a run-time that is $\mathcal{O}[\kappa \log(1/\epsilon) \mathrm{polylog}(M, N)]$. Given the prevalence of matrix inversion as a subroutine in classical algorithms and the $\mathrm{poly}(M, N) \to \mathrm{polylog}(M, N)$ improvement for QLSP, it is no surprise that QLSP solvers have been used as components in proposed quantum algorithms targeting various problems, including differential equations [73–75].

While the potential applications of QLSP solvers may seem vast at face value, there are many technical barriers to achieving end-to-end speedups for classical problems. For example, QLSP solvers require efficient access to entries of A and $\boldsymbol{b}$ and provide an exponential speedup only when the condition number κ is $\mathcal{O}[\mathrm{polylog}(N, M)]$. Moreover, the output of a QLSP solver is a state $|\boldsymbol{y}\rangle$ that is stored in quantum memory. Classical state readout, or the inference of the full classical vector $\boldsymbol{y}$ from the quantum state $|\boldsymbol{y}\rangle$, wipes out any quantum speedup of a QLSP solver. For this reason, a QLSP solver should not be thought of as a direct solver for a classical linear system of equations but rather as a routine to prepare the quantum state $|\boldsymbol{y}\rangle$. The state $|\boldsymbol{y}\rangle$ may, in turn, be used to measure specific observables of interest or as an input to other quantum subroutines; see Refs. [74,76–78] for examples of classical problems that benefit from this use of a QLSP.

We consider the original HHL algorithm in this work and restrict to the case of Hermitian matrices A. Though asymptotically suboptimal, the original HHL algorithm (a) still uses important quantum subroutines like the quantum Fourier transform and quantum phase estimation (QPE) [79] and (b) can be distilled into a small enough circuit for execution on current quantum hardware, unlike more sophisticated QLSP solvers.

At a high level, the HHL algorithm is as follows [see also Fig. 7(a) for a sketch].

(1) *Initialization*—Prepare the state $|\boldsymbol{b}\rangle = \boldsymbol{b}/\sqrt{\boldsymbol{b}^\dagger \boldsymbol{b}}$.

(2) *Eigenvalue estimation*—Denoting the eigenvalues of A by λ_j, the corresponding eigenvectors

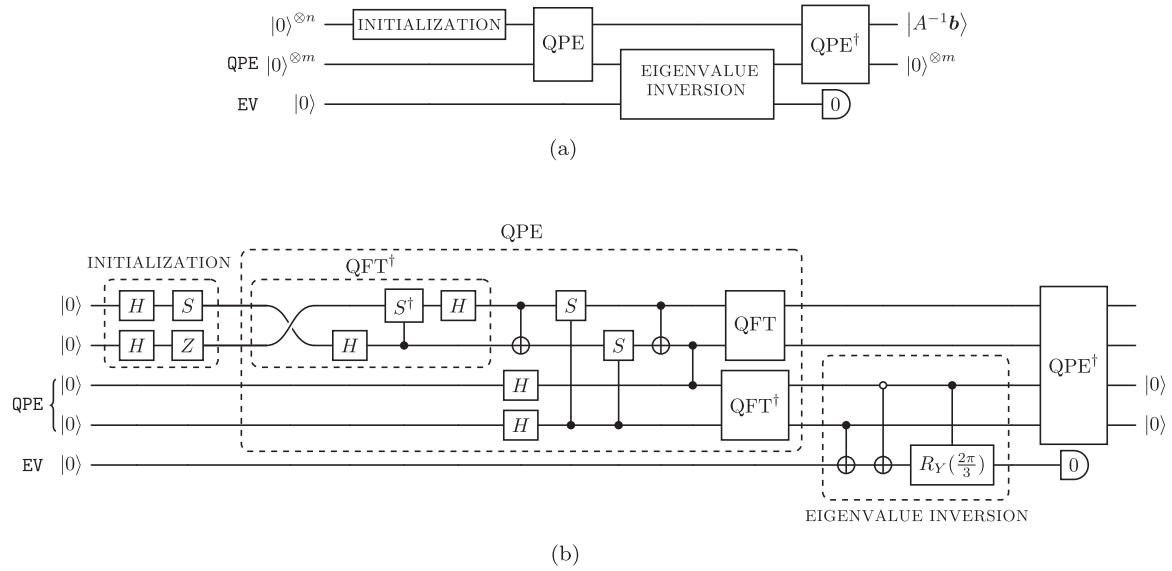


FIG. 7. Logical circuits for the HHL algorithm. The HHL algorithm includes an explicit measurement and postselection on a $|0\rangle$ state of the "eigenvalue" (EV) qubit. In the absence of errors, the remaining (QPE) ancillas are guaranteed to end in $|0\rangle$ states, which can be leveraged to perform logical error detection. (a) Sketch of the HHL algorithm for an $N \times N$ matrix A. Here $n = \lceil \log 2N \rceil$, and the choice of m limits the achievable precision of eigenvalue estimation and inversion. The last gate on the bottom qubit indicates postselection on a 0 measurement outcome. (b) Small-scale implementation of the HHL algorithm for the Poisson equation, with $m = n = 2$.

by $|e_j\rangle$, and defining $b_j = \langle e_j|\boldsymbol{b}\rangle$, implement the transformation

$$|\boldsymbol{b}\rangle = \sum_j b_j|e_j\rangle \to \sum_j b_j|e_j\rangle \otimes |\lambda_j\rangle, \tag{7}$$

which can be done using QPE [79].

(3) *Eigenvalue inversion*—Letting $\tilde{\lambda}_j^{-1} = C/\lambda_j$ with some constant C for which all $\tilde{\lambda}_j^{-1} \le 1$ (and defining $\tilde{\lambda}_j^{-1} = 0$ whenever $\lambda_j = 0$), coherently implement the transformation

$$|\lambda_j\rangle \to |\lambda_j\rangle \otimes \left[\tilde{\lambda}_j^{-1}|0\rangle + \sqrt{1-\tilde{\lambda}_j^{-2}}|1\rangle\right]. \tag{8}$$

(4) *Eigenvalue uncomputation*—Run the inverse of step (2) to uncompute the eigenvalue register, arriving at the state

$$\sum_j b_j|e_j\rangle \otimes \left[\tilde{\lambda}_j^{-1}|0\rangle + \sqrt{1-\tilde{\lambda}_j^{-2}}|1\rangle\right]. \tag{9}$$

(5) *Postselection*—Measure the last qubit register and postselect on a measurement outcome of 0, thereby preparing the state

$$|A^{-1}\boldsymbol{b}\rangle \propto \sum_j \frac{b_j}{\lambda_j}|e_j\rangle, \tag{10}$$

which solves the QLSP induced by $(A, \boldsymbol{b})$.

The details of step (1) depend on the access model for $\boldsymbol{b}$. Steps (2) and (4) are implemented by well-studied QPE circuits (Chap. 5 in Ref. [80]) (see also Fig. 19 in Appendix D). Step (3) typically involves quantum arithmetic to map $|\tilde{\lambda}^{-1}\rangle \to |\eta\rangle = |\arcsin\tilde{\lambda}^{-1}\rangle$ (Sec. VI in Ref. [81]) and then coherently apply (for example) a rotation $R_Y(2\eta)$ to an ancilla qubit that is initially in $|1\rangle$. More generally, every step above can be relaxed to implement a sufficiently close approximation of the desired transformation.

In full generality, HHL relies on quantum arithmetic and Hamiltonian simulation subroutines that may require more advanced quantum computers than are currently available. As a benchmark, we therefore consider using HHL to solve a structured toy problem that is amenable to near-term implementation. Specifically, we consider using HHL to solve the QLSP for the Poisson equation $\nabla^2\boldsymbol{x} = \boldsymbol{b}$ on a translationally invariant one-dimensional lattice of $L = 4$ points with periodic boundary conditions, for which

$$\nabla^2 = S_+ + S_- - 2\mathbb{1}, \tag{11}$$

where $S_\pm = \sum_{j=0}^{L-1} |j \pm 1 \bmod L\rangle\langle j|$ is a shift matrix and $\mathbb{1}$ is the identity matrix. We construct a circuit implementing

HHL for this QLSP with $\boldsymbol{b} = (1, -1, i, -i)$ in Appendix D, altogether arriving at the circuit in Fig. 7(b).

C. Logical circuit compilation

QAOA circuits rely on continuously parametrized rotation gates, which cannot be implemented transversally due to the Eastin-Knill theorem [82]. We, therefore, compile QAOA circuits to a Clifford + T gate set by (a) mapping all rotations of the form $e^{-i\theta Z^{\otimes k}/2}$ to single-qubit $R_Z(\theta) = e^{-i\theta Z/2}$ gates with CNOT ladders and (b) using TRASYN [83] to decompose every $R_Z(\theta)$ gate into a sequence of Clifford + T gates approximating it. TRASYN is a compiler that, given a Clifford + R_Z circuit and a T-gate infidelity, returns a Clifford + T circuit that tries to balance both synthesis error (that is, the fidelity of the decomposed circuit with respect to the original circuit, in the absence of noise) and gate error (determined by the number of T gates). In principle, a detailed and precise noise model of quantum hardware may be used to find an optimal decomposition of a Clifford + R_Z circuit into Clifford + T. In practice, TRASYN uses a Pauli noise model on the logical level, which does not account for how physical errors (including coherent and idling errors particular to the QCCD architecture) affect logical performance. We, therefore, tune the T-gate infidelity provided to TRASYN in order to arrive at circuits with the T-gate counts in Sec. IV. An interesting direction for future work is to investigate noise-aware logical circuit compilation that is faithful to the noise profile of a QCCD architecture.

In the case of the HHL circuit in Fig. 7(b), each controlled-S gate can be decomposed exactly into Clifford + T using three T gates, while the controlled-$R_Y[(2\pi)/3]$ can be approximated to a channel fidelity of approximately 0.966 using only two T gates (see Appendix D).

After an initial decomposition into a Clifford + T gate set, we further compile all logical benchmarking circuits by cycling through rounds of the following:

(1) merging single-qubit gates (commuting them past two-qubit gates whenever possible),
(2) the GreedyPauliSimp compilation pass in PYTKET [84], which implements the Pauli frame graph and Clifford circuit synthesis methods from Refs. [85,86], and
(3) the SimplifyInitial and SimplifyMeasured compilation passes in PYTKET to simplify gates at the beginning and end of the circuit, which act on known initial states or precede measurements.

This cycle of compilation passes is repeated until either the circuit reaches a fixed point or ten compilation cycles are applied, at which point the representation of the circuit throughout compilation that had the fewest T gates (which is typically—but not always—the final circuit) is taken to be the compiled circuit. As a final optimization, we identify cases in which a $|0\rangle$ state is acted on by single-qubit Clifford gates followed by a T gate and replace such cases by an initial $|T\rangle$ state followed by an appropriate sequence of Clifford gates, thereby eliminating a T-gate injection. The compiled logical HHL circuit that we execute on Helios is provided in Fig. 18 in Appendix D.

IV. RESULTS

We now benchmark components used for fault-tolerant quantum computing with the Steane code, before using the best methods identified in our component benchmarks for application-level benchmarking of small-scale quantum algorithms. For brevity, from here onward, we drop the overline notation for logical operators, $\bar{O} \to O$, as our discussions concern only logical degrees of freedom unless explicitly stated otherwise.

A. Component benchmarking

1. $|H\rangle$-state preparation

We begin by benchmarking the two repeat-until-success protocols in Fig. 3 for preparing an $|H\rangle$ state. We use a round of flag-fault-tolerant syndrome measurement for QED in the method of Fig. 3(a). We then measure logical Z and X operators, whose expectation values determine the fidelity

$$\mathcal{F}_{|H\rangle} = \frac{1}{2}\left[1 + \frac{\langle Z\rangle + \langle X\rangle}{\sqrt{2}}\right]. \tag{12}$$

Each expectation value is computed using the H2-1E emulator with the H2-1 noise model [6]. The H2-1E emulator is a classical simulator configured with the calibrated H2-1 device noise model and compiler stack [27,87], which captures the dominant error channels of the device, including a two-qubit gate fault probability of 1.05×10^{-3} (the largest gate error), a single-qubit gate fault probability of 1.9×10^{-5}, measurement errors of the order of 10^{-3}, coherent and incoherent memory errors (informed by ion transport schedules), leakage through spontaneous emission, and crosstalk. Coherent memory errors are mitigated by dynamical decoupling pulses that the H2-1 compiler applies on both the emulator and hardware, and errors from ion transport enter through the associated idling and dephasing. Because the circuits used for our H2-1E benchmarks are small, these simulations use a full state vector simulation. We refer the reader to Appendix D in Ref. [27] and Ref. [87] for a full specification of the noise model.

Figure 8 shows, for different protocols to prepare a logical $|H\rangle$ state, the fidelity $\mathcal{F}_{|H\rangle}$ and discard rate (that is, the probability of a reset and restart due to a nonzero measurement outcome) as a function of a multiplicative factor α by which all error rates of the H2-1 noise model are scaled up or down on the H2-1E emulator. Varying α probes how each protocol would perform as hardware error

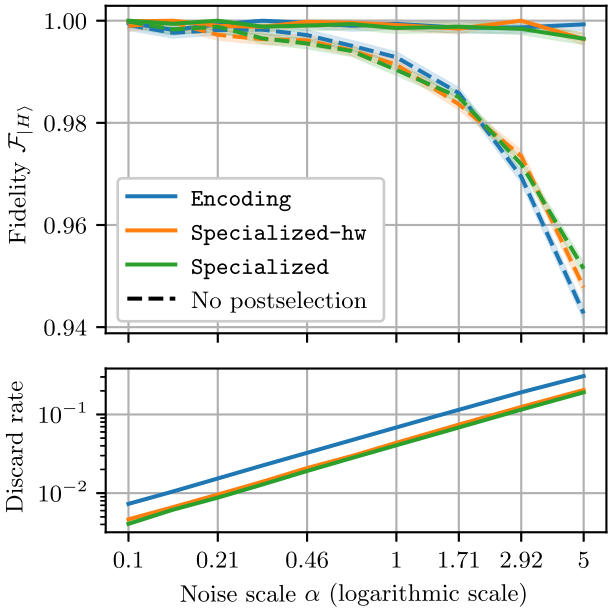


FIG. 8. Simulated state fidelity and discard rate (i.e., the probability of a reset and restart) for different $|H\rangle$-state-preparation protocols as a function of a multiplicative factor α by which all error rates are scaled up or down in the H2-1E emulator with the H2-1 noise model [6,87]. For reference, $\alpha = 1$ corresponds to a probability $p = 1.05 \times 10^{-3}$ of an error after a two-qubit gate. Here, encoding and specialized correspond to the protocols in Fig. 3, while specialized-hw corresponds to the circuit in Fig. 16 in Appendix B, which is used in all hardware experiments in this work. Fidelity is computed using Eq. (12), for which each expectation value is obtained from 78 000 shots. Shaded regions indicate 68% bootstrap confidence intervals for the fidelity (after maximum-likelihood corrections [88]) and standard errors on the mean for the discard rate. After postselection, all $|H\rangle$-state-preparation methods have logical fidelities that are statistically indistinguishable from 1, but the specialized circuits have roughly half the discard rate.

rates decrease ($\alpha < 1$) or increase ($\alpha > 1$), with $\alpha = 1$ corresponding to the calibrated device. Since the expectation values $\langle Z\rangle$ and $\langle X\rangle$ are obtained from independent measurements, sampling error (shot noise) may cause their values to correspond to an unphysical state of a qubit. We, therefore, apply maximum-likelihood corrections to these expectation values [88], which amounts to rescaling the Bloch vector $\boldsymbol{v} = (\langle Z\rangle, \langle X\rangle)$ to $\boldsymbol{v}/\|\boldsymbol{v}\|$ whenever the norm $\|\boldsymbol{v}\|^2 = \langle Z\rangle^2 + \langle X\rangle^2 > 1$.

Overall, we find that fidelities of all protocols considered in Fig. 8 are statistically indistinguishable at the sampled 78 000 shots per expectation value. However, the specialized $|H\rangle$-state circuits [Figs. 3(b) and 16] have roughly half the discard rate of the direct encoding method that was used in previous work [22,24,26] [Fig. 3(a)]. We note that these data do not capture additional benefits of the specialized circuits: the reduction in circuit run-time and the reduction of memory error on idling qubits. The two-qubit gate depth of the specialized circuits is 9, whereas the encoding circuit (with QED via direct measurement of all stabilizers) has a two-qubit gate depth of 25. Since the run-time of a circuit is approximately linear in two-qubit gate depth, specialized circuits reduce the run-time of a magic-state subroutine, potentially reducing both idling errors for other qubits and the overall run-time of the quantum computation, which is typically bottlenecked by the run-time of such subroutines.

2. *T* gates

We now benchmark variants of the Steane-encoded T gate on H2-1. To this end, we run a Ramsey protocol wherein we prepare a logical $|+\rangle$ state, apply n_T sequential T gates (where n_T is a multiple of 4), and perform a logical X-basis measurement. The Ramsey fidelity $\mathcal{F}_T$ of the prepared state with respect to the target state is then the fraction of times that the state is found to be $T^{n_T}|+\rangle \in \{|+\rangle, |-\rangle\}$ upon measurement. Following the procedure proposed in Ref. [89] (and used in Ref. [26]), our T-gate benchmarks twirl $|T\rangle$ states by appending SX to every $|T\rangle$-state-preparation gadget with probability $1/2$ (determined online with a pseudorandom number generator).

Figure 9 shows Ramsey fidelities obtained on H2-1 using logical T gates that are applied with both T-bare and T-swap gadgets in Fig. 4, for which logical $|T\rangle = HS|H\rangle$ states are prepared using the specialized circuit for $|H\rangle$ in Fig. 16 in Appendix B [an older version of the circuit in Fig. 3(b)]. For reference, we also plot Ramsey fidelities

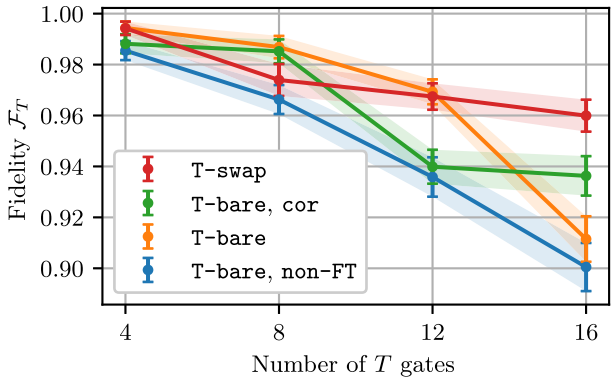


FIG. 9. Experimental Ramsey fidelities obtained from H2-1 using T gates that are applied with the gadgets in Figs. 4(a) (T-bare) and 4(b) (T-swap). In the case of non-FT, logical $|T\rangle$ states are prepared non-fault-tolerantly using the encoding subcircuit in Fig. 3(a), without a logical H measurement or QED. In all other cases, logical $|T\rangle$ states are prepared fault tolerantly using the circuit in Fig. 16 in Appendix B [an older variant of Fig. 3(b)]. Experiments labeled cor apply active bit-flip corrections after every logical T gate based on a measured syndrome. Every data point is obtained from 1000 shots. Error bars indicate 68% Wilson confidence intervals. Interpolating lines and shaded regions are provided as a guide to the eye. Fault-tolerant gadgets outperform the non-fault-tolerant gadget used in prior work, and the T-swap generally performs best.

obtained in Ref. [26] by preparing $|T\rangle$ states non-fault-tolerantly via the encoding subcircuit in Fig. 3(a), without a logical measurement of H or QED, as well as fidelities that we obtain identically on H2-2.

Because of the limited support for conditional ion transport on H2-1, the results in Fig. 9 are postselected on the successful preparation of logical $|H\rangle$ states, for which all ancilla measurement outcomes are 0. This postselection amounts to neglecting idling errors that would otherwise accumulate whenever the $|H\rangle$-state-preparation protocol fails and restarts. We allow for the conditional reset and restart of $|H\rangle$-state-preparation protocols in our application-level benchmarking on Helios in Sec. IV.

The contribution of T-gate errors to the Ramsey fidelity $\mathcal{F}_T(n_T)$ grows with the sequence length n_T, while the contribution of state preparation and measurement error is independent of n_T. Ramsey sequences can, therefore, be used to estimate the infidelity of a single T gate as $1-\mathcal{F}_T(n_T)^{1/n_T}$. The Ramsey fidelity of 0.960(6) for T-swap at $16T$ gates thereby corresponds to a T-gate infidelity of $2.6(4)\times10^{-3}$. For comparison, an identical calculation from prior work (Fig. 5, T method 2 in Ref. [26]) yields a logical T-gate infidelity of 14×10^{-3}.

The T-gate infidelity can also be computed by fitting experimental data on $\mathcal{F}_T(n_T)$ to a single-parameter logical dephasing model, DepModel (see Appendix E for details), which yields a T-gate infidelity of $2.7(4)\times10^{-3}$ for the T-swap gate. However, DepModel predicts an exponential decay of the Ramsey fidelity with respect to the number of T gates, which is a poor fit to the experimental data in Fig. 9. The observed nonexponential behavior reflects a violation of the assumption in DepModel that logical T gates cause independent and identically distributed (IID) logical errors (though we note that other effects, such as coherent logical errors and non-Markovian noise [90], may also contribute to a nonexponential decay of fidelity). In contrast, the Ramsey experiments in Fig. 9 and Ref. [26] allow data-qubit (physical) Z errors to accumulate throughout the entire circuit. The question of whether a logical error has occurred, therefore, depends on how pairs of single-qubit Z errors are correlated *across* different T-gate gadgets. As a consequence, the overall probability of a logical error acquires an additional term that is quadratic in n_T, leading to a Gaussian decay of the Ramsey fidelity (see Appendix F).

The IID assumption of the DepModel can be enforced by the use of extended rectangles [40], which amounts to inserting a QEC cycle between every pair of T gates. However, adding such a high density of QEC cycles may decrease the Ramsey fidelity due to increased opportunities for gate, measurement, and idling errors. We illustrate this effect in Fig. 10, which shows the decay of fidelity in Ramsey protocols with and without QEC cycles, performed using the Selene simulator of Guppy programs [91,92]. Notably, adding a QEC cycle between every pair of logical

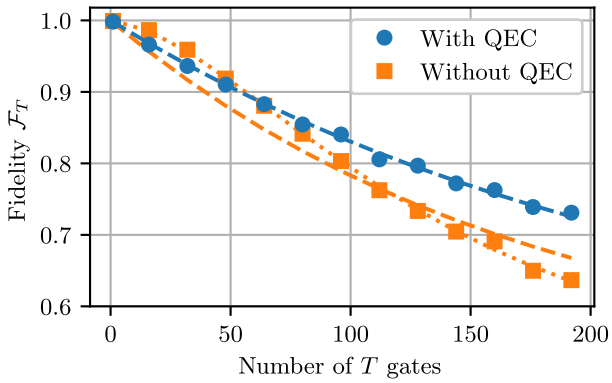


FIG. 10. Simulated Ramsey fidelities as a function of the number of T gates with and without a QEC cycle after every T gate. Dashed lines show fits to a single-parameter exponential decay model (Appendix E), while the dotted line shows a fit to the two-parameter decay model that accounts for the effect of phase error accumulation throughout a Ramsey sequence without intermediate QEC cycles (Appendix F). Adding QEC cycles recovers exponential decay of fidelity, but at the cost of decreased Ramsey fidelity for $\lesssim$64 T gates, beyond which QEC improves the Ramsey fidelity. The results here were obtained from stabilizer simulations of a Clifford proxy for the Ramsey sequence, using the Selene simulator of Guppy programs with two-qubit depolarizing gate rates of 2×10^{-3}, measurement error rates of 10^{-3}, an RUS limit of 3, and the T-swap protocol.

T gates recovers exponential decay behavior but decreases the overall Ramsey fidelity when the total number of T gates is $\lesssim$64. This crossover follows directly from the Gaussian decay model of Appendix F: a QEC cycle after every T gate corrects accumulating Z errors but incurs a fixed per-gate cost set by the infidelity of a QEC cycle. QEC, therefore, helps only for long sequences in which the infidelity due to phase error accumulation exceeds the infidelity of a QEC cycle. The subtle technical challenge of quantifying—or even defining—the fidelity of an isolated logical T gate motivates a shift in focus beyond component benchmarks toward application-level hardware benchmarking. Nonetheless, we remark that fitting the T-swap data in Fig. 9 to the nonexponential decay model in Appendix F yields the same infidelity as the DepModel, of $1-\mathcal{F}_T(1)=2.7(4)\times10^{-3}$.

3. Quantum error correction

We now benchmark different strategies for performing QEC. Letting $\mathcal{Q}$ denote the quantum channel that one (noisy) QEC cycle applies to a logical qubit, the channel fidelity of n QEC cycles with respect to the identity channel $\mathcal{I}$ can be mathematically obtained by applying $\mathcal{Q}^n$ to one qubit of a maximally entangled Bell pair $|\phi\rangle=(1/\sqrt{2})(|00\rangle+|11\rangle)$ and measuring the fidelity of the resulting state with the initial Bell state, altogether arriving at the entanglement fidelity [93]

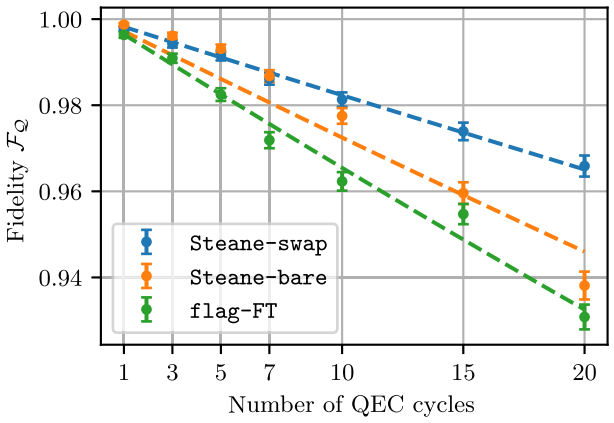


FIG. 11. Simulated entanglement fidelity as a function of the number of QEC cycles performed using different QEC gadgets: flag-fault-tolerant QEC (flag-FT) and two types of Steane-type QEC (Steane-bare and Steane-swap). For the Steane-type methods, a single QEC cycle consists of two rounds of error correction, one each for X-type and Z-type errors. QEC cycles were simulated on the H2-1E emulator with 2000 shots for each of the six initial states used to obtain a single entanglement fidelity. Dashed lines show a fit to a single-parameter depolarizing noise model. Error bars indicate standard errors on the mean.

$$\mathcal{F}_{\mathcal{Q}}(n) = \langle\phi|(\mathcal{I}\otimes\mathcal{Q}^n)(\phi)|\phi\rangle, \tag{13}$$

where $\phi = |\phi\rangle\langle\phi|$. Letting $\Pi_{P,b}$ denote the projector onto the eigenstate of the Pauli operator $P\in\{X,Y,Z\}$ with eigenvalue $(-1)^b$, such that, for example, $\Pi_{Z,0} = |0\rangle\langle 0|$, $\Pi_{Z,1} = |1\rangle\langle 1|$, and $\Pi_{X,0} = |+\rangle\langle +|$, the entanglement fidelity can equivalently be written as (see Appendix G)

$$\mathcal{F}_{\mathcal{Q}}(n) = \frac{1}{4}\sum_{\substack{P\in\{X,Y,Z\}\\ b\in\{0,1\}}} \mathrm{Tr}[\mathcal{Q}^n(\Pi_{P,b})\Pi_{P,b}] - \frac{1}{2}. \tag{14}$$

In words, $\mathrm{Tr}[\mathcal{Q}^n(\Pi_{P,b})\Pi_{P,b}]$ is the survival probability of an initial state $\Pi_{P,b}$ after n applications of the channel $\mathcal{Q}$, that is, the probability that the state is still found to be $\Pi_{P,b}$ upon measurement in the P basis. The QEC fidelity $\mathcal{F}_{\mathcal{Q}}(n)$ can, therefore, be computed by preparing each of six Pauli eigenstates $\Pi_{P,b}$, applying repeated rounds of QEC, measuring survival probabilities $\mathrm{Tr}[\mathcal{Q}^n(\Pi_{P,b})\Pi_{P,b}]$, and combining these probabilities according to Eq. (14).

Figure 11 shows entanglement fidelities obtained from the H2-1E hardware emulator for each of the flag-fault-tolerant (flag-FT), Steane-bare, and Steane-swap QEC protocols in Sec. II E. We note that the limited support for conditional ion transport in the compiler for H2-1 (and H2-1E) disadvantages the flag-fault-tolerant QEC strategy, for which ions are transported (albeit without gates or ion cooling) for a secondary round of QED even if the first round of flagged stabilizer measurements yields a trivial syndrome. Altogether, we conclude from Fig. 11 that the Steane-swap protocol outperforms Steane-bare and flag-FT, though future improvements to the H2-1 compiler may improve the fidelity of flag-FT.

A single-parameter logical depolarizing model in which each QEC cycle $\mathcal{Q}$ has probability p to replace the logical state of a qubit with the maximally mixed state predicts that these fidelities should decay as (see Appendix G)

$$\mathcal{F}_{\mathcal{Q}}(n) = \frac{1}{4} + \frac{3}{4}(1-p)^n. \tag{15}$$

Fitting the simulation data in Fig. 11 to this logical depolarizing noise model, we find entanglement infidelities [defined by $\epsilon = 1 - \mathcal{F}_{\mathcal{Q}}(1) = \frac{3}{4}p$] of $3.53(12)\times 10^{-3}$ for the flag-FT protocol and $1.79(3)\times 10^{-3}$ for the Steane-swap protocol, where uncertainties in the final digits reflect the standard error from the fit. The principled inference of an entanglement infidelity for the Steane-bare protocol is obstructed by its nonexponential decay in Fig. 11. We attribute this nonexponential decay to the accumulation of leakage errors, analogous to the accumulation of Z errors in a Ramsey sequence (see Fig. 10): because the Steane-bare gadget retains the logical state on the same physical data qubits from cycle to cycle, leakage accumulates without being removed. The Steane-swap gadget, by contrast, teleports the logical state onto fresh qubits every cycle and thereby mitigates leakage [see Fig. 4(d)], consistent with its clean exponential decay. The flag-FT experiment also suffers from the accumulation of leakage errors but additionally exhibits irregular and non-Markovian behavior due to the intrinsically dynamic nature of the flag-FT QEC cycle (which uses measurement outcomes to make real-time choices about executing subroutines). These observations do not impact our primary conclusion that Steane-swap is the preferred method for performing active QEC.

B. Application benchmarking

Having benchmarked the components that are used to run error-corrected quantum algorithms, we now consider application-level benchmarks. To this end, we construct various QAOA and HHL circuits, compile these circuits into a Clifford + T gate set (as discussed in Sec. III C and Appendix D), and expand each compiled logical circuit into a physical circuit using Steane-code gadgets. Specifically, we use the specialized $|H\rangle$-state-preparation circuit in Fig. 16 in Appendix B [an older version of the circuit in Fig. 3(b)], the T-swap variant of the T gate in Fig. 4(b), and (when applicable) the Steane-swap variant of a QEC cycle in Fig. 4(d). In some cases, we additionally consider varying the RUS limit.

We execute logical circuits on H2-2 and Helios. While both of these devices support real-time classical feedback, such as conditioning a quantum gate on the outcome of a preceding measurement, H2-2 programs must nonetheless be compiled into an intermediate representation of serialized instructions prior to execution. In contrast, Helios

supports a real-time engine that consumes Guppy [91,92] programs, which allow for nonlinear control flow, such as conditional branching and FOR loops. Moreover, the software stack for Helios allows for real-time dynamic allocation of physical qubits, whereas executing circuits on H2-2 requires allocating all physical qubits at compile time. In particular, to execute our circuits on H2-2 (which has 56 physical qubits), we allocate a fixed choice of seven qubits per Steane code block (including one ancilla code block for T-gate teleportation and Steane QEC) and, additionally, six physical qubits as ancillas that are shared by the Steane code blocks.

1. QAOA

We first consider QAOA circuits for the LABS problem with binary sequences of lengths $N = 5$, 6 (see Sec. III A 1). Figure 12 shows the measured QAOA success probability (that is, the probability of measuring the optimal solution to a LABS instance) with various choices of the device, QAOA depth p, T-gate count n_T, and RUS limit r (see Tables I and II in Appendix H for additional experimental data). While the number of qubits and T gates in these logical circuits are modest, we emphasize that the corresponding physical circuits are considerably larger, involving up to 55 qubits and several hundred two-qubit gates (see Tables I–IV). Remarkably, for both the $N = 5$ and $N = 6$ LABS instances, we observe that increasing the logical T-gate budget improves QAOA performance at depth $p = 2$ (green arrows in Fig. 12), despite a considerable increase in physical circuit complexity and two-qubit gate count (399 → 551 for $N = 5$ and 420 → 686 for $N = 6$ when $r = 1$). We also observe on H2-2 that for $N = 5$ the QAOA success probability can be increased by increasing the QAOA depth $p: 1 \to 2$ at a fixed logical T-gate budget of $n_T = 9$. Increasing QAOA depth similarly improves the performance for $N = 6$; however, in this case the $p = 2$ logical circuit is Clifford, making the physical circuit for $p = 2$ slightly smaller than that for $p = 1$.

Where data are available, we observe that increasing the RUS limit r slightly decreases the QAOA success probability on H2-2 and has no meaningful impact on the success probability on Helios. However, increasing the RUS limit to $r = 2$ drastically reduces the discard rate of hardware executions, or the probability with which a program must terminate and restart due to a corrupted subroutine, to nearly zero. This improvement highlights the importance of dynamic repeat-until-success protocols for making fault-tolerant quantum computation scalable. We remark that the benefit of repeat-until-success on Helios is observed

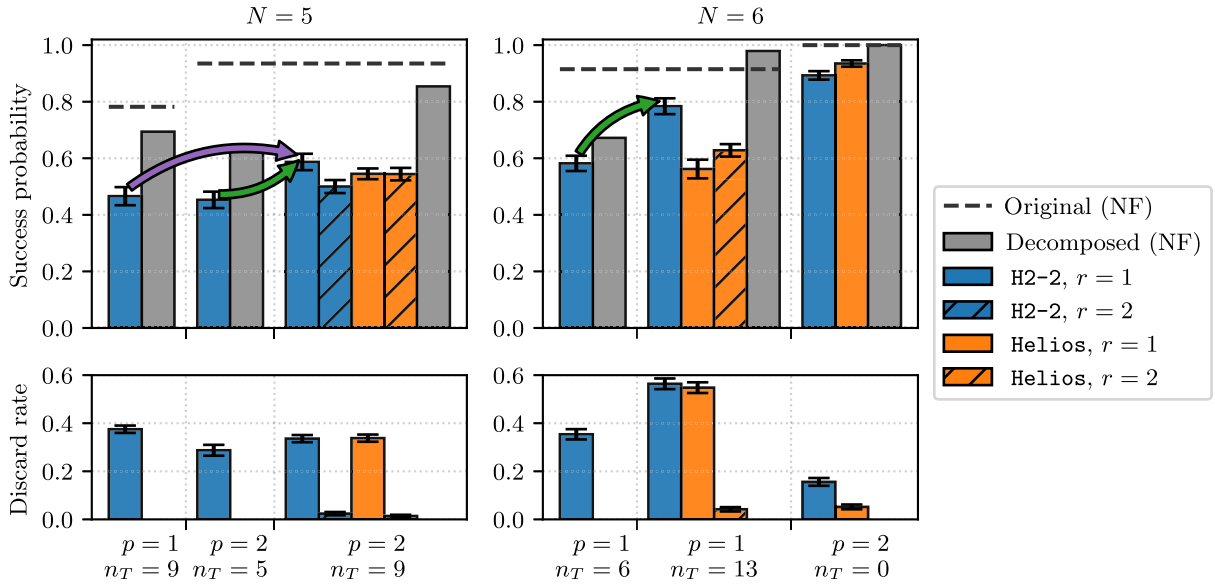


FIG. 12. Probability of measuring the optimal solution to the LABS problem for $N = 5$, 6 bits. Data are grouped by the number of QAOA layers, p, and the total number of T gates in a compiled circuit, n_T. Dashed lines indicate the success probability for exact, noise-free (NF) QAOA circuits with optimal parameters at a given QAOA depth p. Gray bars show the noise-free success probability after decomposition to a Clifford + T gate set [which exceeds the undecomposed circuit at $(N, p, n_T) = (6, 1, 13)$ due to a finite-size effect; note that the decomposed circuit is not limited to the QAOA ansatz structure]. Orange and blue bars show, respectively, the success probability when running on H2-2 and Helios. Solid and hatched bars correspond to RUS limits of $r = 1$ and $r = 2$, respectively. Bottom panels show the fraction of hardware runs that are discarded because a subroutine exceeded the RUS limit. Error bars indicate standard errors on the mean. Increasing p or decomposing to a larger number of T gates increases the success probability on H2-2 (purple and green arrows), indicating that the benefits of finer decomposition outweigh the increased errors from running a larger logical circuit. Increasing the RUS limit r from 1 to 2 slightly decreases the success probability on H2-2 but does not significantly affect the success probability on Helios. However, increasing the RUS limit to $r = 2$ reduces the program discard rate to nearly zero.

despite the lack of dynamical decoupling in the Helios compiler at the time of these experiments. The addition of dynamical decoupling, as well as other compiler improvements, can therefore be expected to yield substantial improvements in future Helios performance.

To assess the scalability of our fault-tolerant QAOA implementation on Helios, we execute logical circuits for LABS problem instances up to $N = 9$ variables and a portfolio optimization problem instance with $N = 12$ variables. For LABS instances with $N > 6$, the preoptimized QAOA parameters in QOKIT yield circuits that, when decomposed into a Clifford + T gate set, exceed the T-gate budget that is executable on Helios within practical coherence times. To identify parameter settings that produce executable circuits while maintaining the performance of the algorithm, we perform a grid search over the two QAOA parameters at depth $p = 1$ and select the best-performing parameters that yield compiled circuits with fewer than $30T$ gates. The results from running these LABS experiments are summarized in Fig. 13, and additional data are provided in Table III in Appendix H. For LABS instances of size up to $N = 8$, we find that the logically encoded circuits yield a probability of success that is significantly greater than a random guess and is, moreover, comparable to the probability of success obtained by unencoded circuit execution. These results suggest the possibility that future hardware upgrades, software-level improvements (such as the addition of dynamical decoupling), and circuit-level decoding may push the performance of error-corrected algorithms beyond breakeven, i.e., with logically encoded circuits outperforming unencoded circuits.

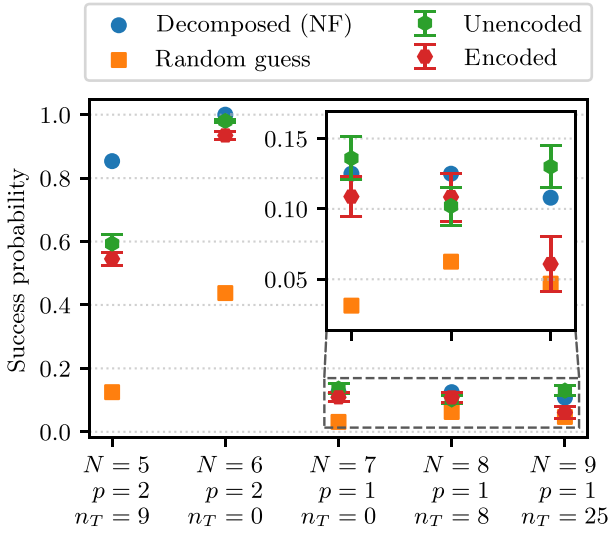


FIG. 13. Success probability for QAOA circuits with QAOA depth p on LABS instances with varying problem size N. "Decomposed (NF)" corresponds to a noise-free circuit that has been decomposed to a Clifford + T gate set (with n_T T gates in the decomposed circuit). "Unencoded" and "encoded" correspond, respectively, to unencoded (physical) and encoded (logical, RUS limit $r = 1$) execution of the same circuit on Helios. "Random guess" shows the probability of success for a randomly chosen bit string. Error bars indicate standard errors on the mean. Decomposed and encoded circuit data for $N = 5$ and $N = 6$ is the same as in Fig. 12. The encoded circuit performs comparably to the unencoded one up through $N = 8$, but performance deteriorates at $N = 9$ due to the large number of T gates.

Finally, we scale QAOA for portfolio optimization to 12 logical qubits, using 97 of the 98 qubits available on the Quantinuum Helios processor. To identify a portfolio optimization instance within the current capabilities of Helios, we generate multiple $N = 12$ problem instances by randomly sampling mean return vectors and covariance matrices from Yahoo Finance data using QOKIT [68]. For each problem instance, we numerically optimize the QAOA parameters and compile the resulting circuits using the procedure described in Sec. III C. We then select an instance whose compiled logical circuits at QAOA depths $p = 1, 2$ have small numbers of T gates. Figure 14 shows a histogram of approximation ratios for the logical bit strings sampled from these circuits on Helios; additional data are provided in Table IV in Appendix H. We remark that these circuits use $7 \times 12 = 84$ physical qubits to encode the $N = 12$ problem variables, seven qubits for a logical ancilla qubit, and additionally six physical ancilla qubits, for a total of 97 physical qubits. The compiled $p = 1$ circuit has 1132 physical two-qubit gates, and the $p = 2$ circuit has 2132 physical two-qubit gates. Although the $p = 2$ circuit performs worse than that with $p = 1$ due to its larger physical gate count, both circuits outperform random guessing ($p = 0$). While this portfolio optimization problem is highly simplified and the circuits are produced in a nonscalable way, our results show that fault-tolerant and

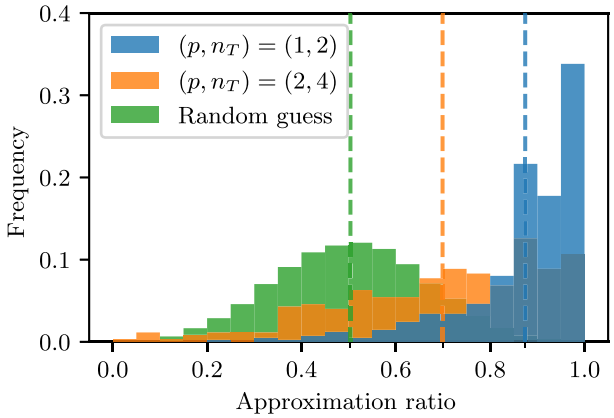


FIG. 14. Histogram of approximation ratios for a portfolio optimization problem instance with $N = 12$ binary variables, obtained by running logical QAOA on Helios with an RUS limit of $r = 1$. Vertical axis values indicate the fraction of outcomes that fall within each bin (of width 0.05). The distribution of approximation ratios for randomly chosen bit strings is provided for reference. Dashed vertical lines indicate distribution means.

error-corrected execution of algorithmic circuits with 12 logical qubits can achieve high fidelity.

2. HHL

For our final application benchmark, we run the HHL circuit in Fig. 7(b) on Helios. After decomposing into a Clifford + T gate set (see Appendix D for details) and postselecting on a measurement of 0 on the last ancilla qubit, this circuit prepares the state

$$|\chi\rangle = \left(\frac{|0\rangle + ie^{i\eta}|1\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right), \tag{16}$$

where $\eta = \arccos\sqrt{8/9} \approx 0.11\pi$. The undecomposed circuit prepares the same state with $\eta = 0$. The fidelity $\mathcal{F}_{|\chi\rangle}(\rho) = |\langle\chi|\rho|\chi\rangle|^2$ of a (possibly mixed) state ρ with respect to $|\chi\rangle$ can be recovered from measurements of the first qubit in the X and Y bases and the second qubit in the X basis. In the absence of errors, the QPE qubits are guaranteed to end in logical $|0\rangle$ states. This fact can be leveraged to perform algorithm-level error detection by measuring the QPE qubits and discarding shots in which the QPE measurement outcomes are nonzero.

Figure 15 shows the fidelities and discard rates of logical HHL circuits executed on Helios with different RUS limits r and numbers of QEC cycles, n_{QEC} (see Fig. 18 in Appendix D for the compiled logical circuit and the locations of QEC cycles as well as Table V in Appendix H for additional experimental data). The physical circuits for these experiments use 48 physical qubits and, at an RUS limit of $r = 1$, have two-qubit gate counts of (677, 749, 821) for $n_{\mathrm{QEC}} = (0, 2, 4)$. We find that the addition of two QEC cycles in the middle of the circuit improves the fidelity of the measured logical state with respect to the expected state in Eq. (16), which can be as high as 0.97 ± 0.02. However, additional QEC cycles can degrade the performance of the algorithm, reflecting that the Steane code is near—but not yet below—the breakeven point for these algorithms. Similarly to our QAOA benchmarking, we find that increasing the RUS limit to $r = 2$ decreases the probability of failure for state-preparation subroutines to nearly zero (see SP discard rates in Fig. 15). Unlike

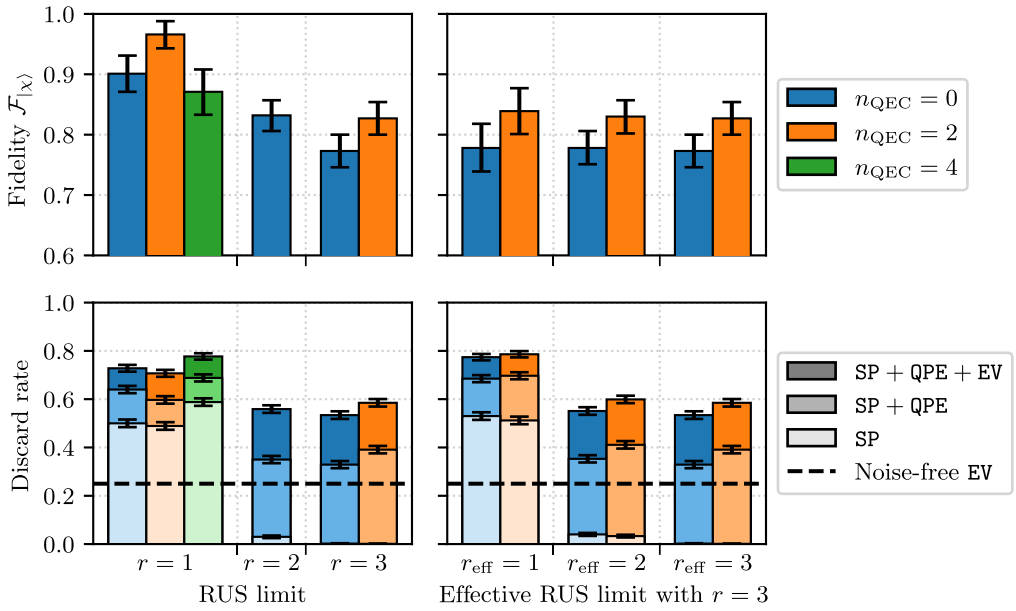


FIG. 15. Target state fidelities and discard rates of logical HHL circuits executed on Helios with varying RUS limits r (left) or when $r = 3$ and every RUS subroutine succeeded in at most r_{eff} attempts (right). Color indicates the number of QEC cycles, n_{QEC}, inserted into the HHL circuit (see Fig. 18 in Appendix D for the compiled circuit and QEC cycle locations). Shade indicates postselection criteria: trivial flags at the end of all state-preparation routines (SP), logical measurement outcomes of 0 on the QPE qubits, and a logical measurement outcome of 0 on the EV qubit. The dashed line in the bottom panels indicates the EV discard rate of the noise-free compiled HHL circuit. Error bars indicate 68% confidence intervals obtained by bootstrapping (top panels) or standard errors on the mean (bottom panels). Adding two QEC cycles ($n_{\mathrm{QEC}} = 2$) improves logical circuit fidelity, but additional QEC cycles can introduce more errors than they mitigate ($n_{\mathrm{QEC}} = 4$), reflecting that the performance of Steane-encoded algorithms is not yet at breakeven. Increasing the RUS limit drops the failure rate of state-preparation subroutines to nearly zero but at a moderate cost to the logical fidelity of the HHL circuit. However, this fidelity penalty depends on the compile-time RUS limit r (top left panel), not the maximal number of attempts actually used at run-time, r_{eff} (top right panel), indicating that the fidelity penalty is a result of compile-time behavior rather than run-time or hardware effects.

the case of QAOA, however, this benefit comes at the cost of reduced logical circuit fidelity for the HHL circuit (see the top left panel in Fig. 15).

To diagnose the reason for a fidelity penalty with higher RUS limits, we examine the data from programs with an RUS limit of $r = 3$ more closely and consider the shots in which all RUS subroutines succeed in at most $r_{\text{eff}} \in \{1, 2, 3\}$ attempts. This analysis acts as an experimental control to isolate effects that occur in hardware from those that occur at the software level. In principle, if a program has an RUS limit of r but all RUS subroutines succeed in at most $r_{\text{eff}} \leq r$ attempts, the results on hardware should be identical to those of a program with an RUS limit of r_{eff}. This observation lets us diagnose the source of the fidelity penalty. If the penalty came from actually executing more state-preparation attempts, it would depend on the maximal number attempts used at run-time, r_{eff}. Instead, we find that the fidelity depends on the compile-time RUS limit r (top left panel in Fig. 15) but is essentially independent of r_{eff} (top right panel). The fidelity penalty is, therefore, attributed to compile-time behavior rather than run-time or hardware effects. We attribute this effect to ion-transport scheduling in the QCCD architecture: each repeat-until-success branch introduces conditional branching logic that enforces constraints on the transport scheduler. Compiling a dynamic program with a larger RUS limit may, therefore, yield a less efficient transport schedule with more ion movement and idling throughout the circuit, even if the extra repeat attempts are never used. Importantly, the compiled circuits remain logically correct, reproducing the intended circuit in noise-free simulation; the RUS limit affects only scheduling efficiency and, hence, idling error. Because this is a scheduling effect and not a miscompilation, it does not undermine our other results, which always compare circuits compiled in the same way; here, it appears only as the modest, explicitly reported fidelity cost of raising the RUS limit. This observation highlights the importance of application-level benchmarking for uncovering behaviors of a quantum computer that isolated component benchmarking may not reveal.

Finally, we observe in Fig. 15 that the probability of obtaining nonzero logical measurement outcomes on the QPE qubits is consistently high (> 0.3), implying a large logical error rate, even in the case of $r \geq 2$ when all state-preparation subroutines succeed with high probability (indicated by a low SP discard rate). We remark that the use of fault-tolerant state-preparation subroutines, which account for the vast number of physical gates in the HHL circuit, suppresses the physical gate errors of $p \sim 10^{-3}$ on Helios to logical error rates of $\mathcal{O}(p^2)$. The large number of physical gates in the HHL circuit, therefore, cannot account for the observed logical error rates. It is unclear whether idling errors can account for the observed logical error rates, though upgrades to the Helios compiler—such as the addition of dynamical decoupling—should mitigate idling errors and provide clarity on their impact. These logical error rates also reflect the need for either extended rectangles (by, for example, adding a QEC cycle on every logical qubit between every pair of gates) [40] or correlated decoding (in the spirit of algorithmic fault tolerance [16,17]) to control the propagation of physical errors throughout the circuit. Without such techniques, physical errors can propagate to logical errors even when using transversal gates and fault-tolerant gadgets (see Appendix A for a toy example). In practice, the construction of extended rectangles incurs substantial quantum computing overheads, whereas correlated decoding can be performed at the software level with modest classical computing overheads (see, for example, Ref. [18]). We, therefore, anticipate that future efforts to fault-tolerantly execute error-corrected quantum algorithms on trapped-ion quantum architectures will require dynamical decoupling and more advanced decoding techniques to further suppress logical error rates.

V. DISCUSSION

We initiate the application-level benchmarking of quantum processors at the logical level with both error correction and fault tolerance, which was previously obstructed by small qubit counts and low fidelities. Our results show that existing trapped-ion quantum computers, namely, Quantinuum System Model H2 and Quantinuum Helios, operate near breakeven for complex algorithmic circuits that are logically encoded into the Steane code, altogether involving dozens of physical qubits and hundreds of gates. Furthermore, our experiments suggest that modest improvements to fault-tolerant circuit gadgets, dynamic program compilation, correlated decoding, or the choice of QEC code used may be sufficient to achieve the algorithmic breakeven point [94] at which fault-tolerantly encoded quantum programs outperform their unencoded counterparts.

QAOA has long been used as a benchmark for physical quantum processors. In the absence of noise, increasing QAOA depth can only improve performance. However, on noisy devices, increasing depth causes a corresponding growth of errors, leading to the existence of an optimal QAOA depth $p_\star$ beyond which performance no longer improves. Implementing N-qubit QAOA with a large $N \cdot p_\star$ was the central goal of the DARPA ONISQ program [95]. Multiple independent benchmarking studies have used this quantity to characterize the capabilities of physical [96–100] or partially fault-tolerant [101,102] devices. In this work, we extend this approach to

benchmarking fault-tolerant error-corrected processors, achieving $N \cdot p_\star = 6 \cdot 2 = 12$.

Targeting application-level benchmarks yields several important insights. First, it provides an overview and synthesis of all concepts and components that must be integrated for fault-tolerant and error-corrected quantum program execution. As devices continue to scale up, it is important to continue identifying and benchmarking alternative methods for each necessary function, such as active QEC or magic-state injection. Second, our application benchmarks demonstrate key capabilities, most notably the ability to suppress errors through active QEC and to reduce discard rates through dynamic (RUS) subroutines, at minimal or no cost to algorithm performance. Finally, these benchmarks reveal gaps and challenges that need to be addressed in future work. The compiled HHL circuit, for example, requires 14 magic states and 28 two-qubit gates, but its logical error rate is substantially worse than the component logical error rates of approximately 10^{-3} would suggest (e.g., a rough estimate of $1 - 0.999^{14} \approx 0.01$ obtained by counting logical T gates or even $1 - 0.999^{14+28} \approx 0.04$ by counting logical T and CNOT gates). Instead, our application benchmarks reveal the importance of factors such as the compilation of dynamic programs, the fault-tolerant *composition* of fault-tolerant gadgets (e.g., via extended rectangles), and correlated or circuit-level decoding. The shift in focus beyond components and error rates as bottlenecks of quantum algorithm performance simultaneously speaks to the maturity of the field and signals the advent of the early-fault-tolerant era for quantum computing.

ACKNOWLEDGMENTS

We thank Matthew DeCross and Kentaro Yamamoto for helpful feedback on a draft of this manuscript. We acknowledge the entire Quantinuum team for innumerable contributions toward the design and operation of the H2 and Helios quantum computers, and we acknowledge Honeywell for fabricating the ion trap used in this experiment. We thank Rob Otter for executive support of this work and invaluable feedback on this project. We thank the technical staff at JPMorganChase's Global Technology Applied Research for their support.

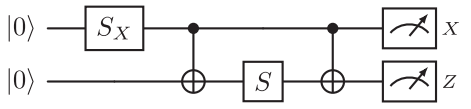
This paper was prepared for informational purposes with contributions from the Global Technology Applied Research center of JPMorgan Chase & Co. This paper is not a product of the Research Department of JPMorgan Chase & Co. or its affiliates. Neither JPMorgan Chase & Co. nor any of its affiliates makes any explicit or implied representation or warranty, including, without limitation, with respect to the completeness, accuracy, or reliability of the information contained herein and the potential legal, compliance, tax, or accounting effects thereof. This document is not intended as investment research or investment advice, or as a recommendation, offer, or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction.

DATA AVAILABILITY

Most data used in this work are provided in Appendix H. Additional data are available from the authors upon reasonable request.

APPENDIX A: LOGICAL ERRORS AND CORRELATED DECODING

Here, we consider a toy example of a logical circuit in which physical error rates of p lead to logical error rates of $\mathcal{O}(p)$, even when encoding into the Steane code using fault-tolerant gadgets:

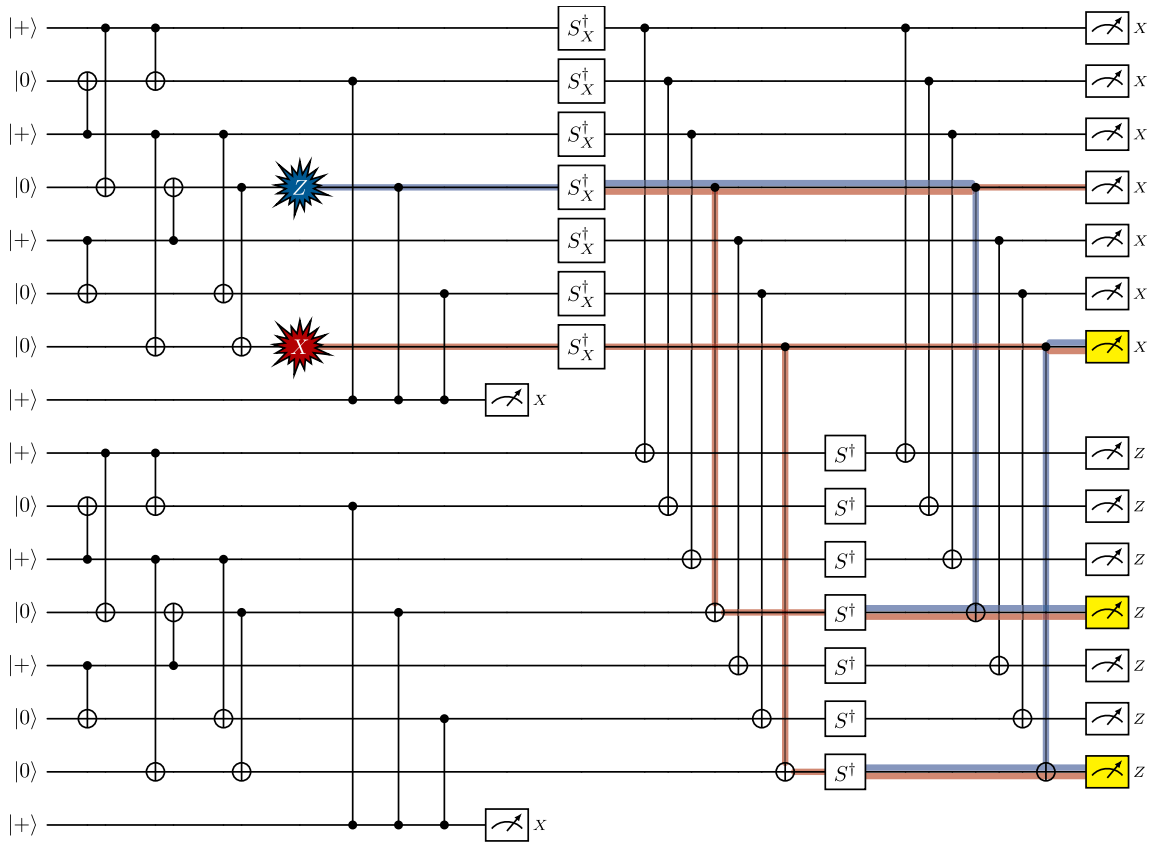


(A1)

The two-qubit state in this circuit transforms as

$$|00\rangle \xrightarrow{S_X} |00\rangle - i|10\rangle \xrightarrow{CX} |00\rangle - i|11\rangle \xrightarrow{S} |00\rangle + |11\rangle \xrightarrow{CX} |00\rangle + |10\rangle = |+\rangle \otimes |0\rangle. \tag{A2}$$

In the absence of errors, both final measurement outcomes should, therefore, be 0. However, for Calderbank-Shor-Steane codes such as the Steane code, fault tolerance allows state-preparation subroutines to end with weight-2 errors of mixed (X and Z) type, such as $X_0 Z_1$. In turn, S-type gates can change the type of one of these errors, for example, with an S_X gate converting $X_0 Z_1 \to X_0 X_1 \cdot Z_1$, and logical CNOT gates can propagate such errors between logical qubits in such a way that naive, uncorrelated decoding leads to a logical error. As a concrete example, consider the propagation of a two-qubit gate error in the $|0\rangle$-state-preparation circuit of the first logical qubit when encoding into the Steane code:



(A3)

While the measurement error on the top logical qubit would be corrected by a naive decoder, the measurement errors on the bottom logical qubit would be decoded into a logical error, which occurs with (at least) the same probability as that of the propagated physical error. This example illustrates that merely substituting physical gates by fault-tolerant gadgets is insufficient for a logical circuit to be fault tolerant and that suppressing logical errors beyond $\mathcal{O}(p)$, therefore, requires additional measures such as the insertion of QEC cycles to construct extended rectangles and the use of more sophisticated decoding strategies.

APPENDIX B: IMPROVED MAGIC-STATE PREPARATION

Here, we discuss the construction of our specialized circuit in Fig. 16 for fault-tolerantly preparing an $|\bar{H}\rangle$ state on hardware. Additionally, we give a slightly improved variant of the circuit using one fewer ancilla and two fewer CNOT gates, which was identified after completing our experiments.

In Ref. [45], the authors give a fault-tolerant preparation circuit for $|\bar{H}\rangle$ that is composed of three parts:

(1) a non-fault-tolerant magic-state-preparation circuit,
(2) a flag fault-tolerant circuit for measuring the logical Hadamard operator, and
(3) a full syndrome extraction cycle.

We follow a similar overall approach but modify steps (1) and (3) to minimize the number of two-qubit gates. We reduce the number of two-qubit gates from 48 in the circuit given in Ref. [45] to 30 for the circuit used in our experiments and then to 28 in our improved version given here.

First we consider step (1), the non-fault-tolerant magic-state-preparation circuit. As discussed in Sec. II D in the main text, our non-fault-tolerant magic-state-preparation circuit is constructed by backpropagating a representative of $\bar{X}$ until its support has weight 2. This allows us to implement the logical rotation $R_{\bar{X}}(\theta) = \exp[-i(\theta/2)\bar{X}]$ with a physical two-qubit gate, $R_{XX}(\theta) = \exp[-i(\theta/2)X \otimes X]$. Next, following the approach of Ref. [103], we flag this subcircuit to detect as many errors as possible at origin. The circuit is not fully bipartite, so it cannot be fully flagged in this way, but many errors can be detected at this early stage.

Moving on to step (3), we describe our final syndrome extraction gadgets. We seek to find a minimal set of stabilizers such that if we measure them, we will detect all errors arising from faults in the circuit. To do this, we propagate all possible Pauli errors occurring after each gate in the circuit to the end of the circuit. Since the circuit is not Clifford, this can produce linear combinations of Pauli strings at the end of the circuit. For each propagated error,

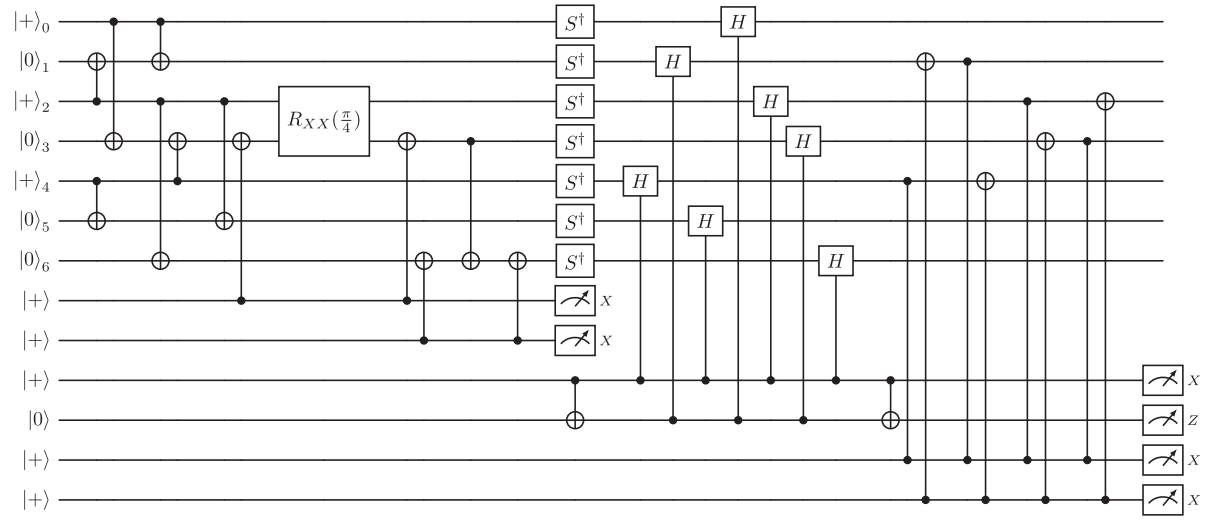


FIG. 16. A circuit for fault-tolerantly preparing logical $|H\rangle$ states with six ancilla qubits, 30 two-qubit gates, and a two-qubit gate depth of 9 (with as-soon-as-possible gate scheduling). This circuit was used for hardware benchmarks in Sec. IV. An improved version of this circuit, discovered after our hardware experiments, is provided in Fig. 3(b), which has one fewer ancilla qubit and two fewer two-qubit gates.

we exhaustively minimize its weight under the action of both the stabilizer group and the logical stabilizer $\bar{H}$. If the propagated error has minimal weight less than 2, then it can be ignored. In the case of linear combinations of Pauli strings, we apply weight reduction and filtering (after possible multiplication with the logical stabilizer $\bar{H}$) to each term of the superposition independently. This procedure gives us a list of problematic Pauli strings that arise from some fault in the circuit and need to be detected. We exhaustively compare this list to the full stabilizer group and find that there is no single element of the stabilizer group that anticommutes with all problematic errors, but there are pairs of weight-4 stabilizers that anticommute with all of them. Out of these pairs, we choose one where the two stabilizers, one X type and one Z type, have the same support, as these can most efficiently be measured together.

Finally, we revisit step (1) and attempt to remove any flag-at-origin gadgets that are no longer needed because they detect errors that are now also detected by the final syndrome measurements. In this case, we are able to remove the majority of them. The circuit in Fig. 16 retains two flags, whereas the circuit in Fig. 3(b) retains only one.

APPENDIX C: FLAG-FAULT-TOLERANT QEC

Here, we describe, for completeness (that is, to provide explicit circuits and protocols for all gadgets used in this work), the flag-fault-tolerant QEC protocol in Ref. [23]. This protocol is adapted slightly to be consistent with the choice of qubit ordering for the Steane code in this work. The protocol is as follows.

(1) Perform a round of XZZ stabilizer measurements using the flagged circuit in Fig. 17(a) and store the

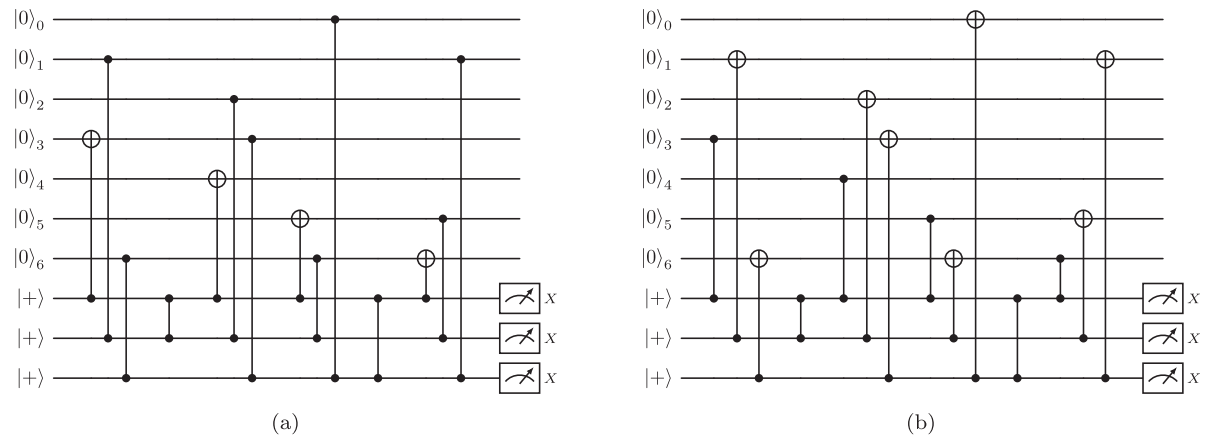


FIG. 17. Subcircuits used for (a) XZZ and (b) ZXX stabilizer measurements in flagged QEC, adapted from Ref. [23].

syndrome measurement outcomes, from top to bottom, to the bits $(x_1, z_2, z_3) \in \{0,1\}^3$.

(2) If $(x_1, z_2, z_3) \neq (0,0,0)$, set $(z_1, x_2, x_3) = (0,0,0)$; otherwise, perform a round of ZXX stabilizer measurements using the flagged circuit in Fig. 17(b) and store the syndrome measurement outcome to the bits $(z_1, x_2, x_3) \in \{0,1\}^3$. If $x_1 = x_2 = x_3 = z_1 = z_2 = z_3 = 0$, stop the protocol without proceeding to the following steps.

(3) Perform a non-fault-tolerant bare-ancilla measurement of all X stabilizer generators, corresponding to rows of the parity check matrix h in Eq. (1), using a circuit analogous to that in Fig. 5. Save the measurement outcome of the stabilizer with support given by the (first, second, third) rows of h to the bits (x_4, x_5, x_6).

(4) Similarly, measure all Z stabilizer generators and save the measurement outcomes to the bits (z_4, z_5, z_6).

(5) Decode the syndromes (x_4, x_5, x_6) and (z_4, z_5, z_6), and perform active error correction as discussed in Sec. II C.

(6) Let

$$E = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}. \tag{C1}$$

If $(x_1, x_2, x_3, x_4, x_5, x_6) \in \text{rows}(E)$, apply a logical Z operator.

(7) If $(z_1, z_2, z_3, z_4, z_5, z_6) \in \text{rows}(E)$, apply a logical X operator.

In words, the last two steps of this protocol correct hook errors that (i) may occur during flagged XZZ and ZXX syndrome readout and (ii) propagate to logical errors. Note that the protocol presented here is technically different from that in Ref. [23], even after accounting for qubit ordering conventions, due to a different choice of stabilizer measurements for (x_4, x_5, x_6) and (z_4, z_5, z_6), which changes the conditions (that is, the matrix E) used to correct hook errors. The protocol outlined here is used for Helios experiments, while the protocol in Ref. [23] (which is implemented in PECOS [48]) was used for H2-2 experiments.

APPENDIX D: DETAILS OF THE HHL IMPLEMENTATION

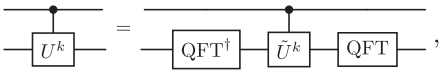
Here, we describe our compilation of the HHL algorithm to a QLSP that is suitable for near-term benchmarking. For reference, the QLSP we consider is that of finding a least-squares solution to the Poisson equation

$$\nabla^2 \boldsymbol{x} = \boldsymbol{b} \tag{D1}$$

for $\boldsymbol{b} = (1, -1, i, -i)$, which corresponds to the initial state $|\boldsymbol{b}\rangle = |+Y\rangle \otimes |-X\rangle$, where $|+Y\rangle = SH|0\rangle$ and $|-X\rangle = ZH|0\rangle$ are Pauli eigenstates. The key components of the HHL circuit that we need to construct are a quantum phase estimation circuit for steps (2) and (4) of the algorithm (as described in Sec. III B) and an eigenvalue inversion circuit for step (3). These components are combined to obtain the circuit in Fig. 7(b). The final logical HHL circuit that we execute on Helios is provided in Fig. 18.

1. Quantum phase estimation

Quantum phase estimation for step (2) of the HHL algorithm is generally implemented to m bits of precision by the circuit in Fig. 19. Judiciously normalizing $A = -\nabla^2/8$, we observe that the eigenvalues of A are $\{0, \frac{1}{2}, \frac{1}{4}\}$, so two bits of precision suffice to compute the eigenvalues of A exactly. Letting $U = e^{i2\pi A}$, we observe that A is diagonalized by the quantum Fourier transform (QFT), which means that



(D2)

where $\tilde{U} = e^{i2\pi\tilde{A}}$ is diagonal and $\tilde{A} = \text{QFT}^\dagger \cdot A \cdot \text{QFT}$. Specifically,

$$\tilde{U} = \text{diag}(1, -i, -1, -i) = CX(S \otimes S)CX, \tag{D3}$$

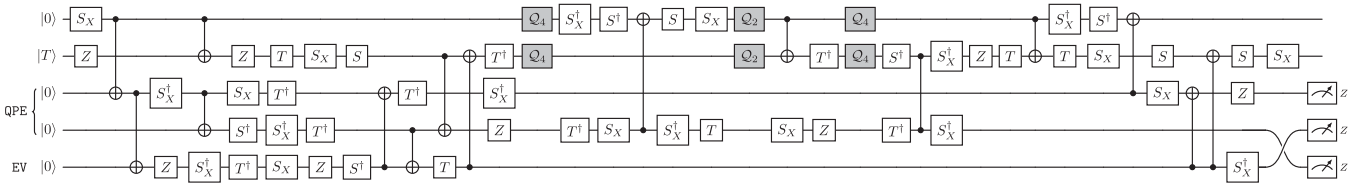


FIG. 18. The logical HHL circuit that we execute on Helios, obtained by compiling the circuit in Fig. 7(b). Gray boxes indicate locations at which QEC cycles are inserted in experiments with a total of $n_{\text{QEC}} = 2$ ($\mathcal{Q}_2$) or $n_{\text{QEC}} = 4$ ($\mathcal{Q}_4$) QEC cycles. An explicit SWAP on the last two qubits is added to track the identity of logical qubits, so that the inputs and outputs to the circuit are ordered identically to Fig. 7(b); in practice, this SWAP is performed virtually by relabeling qubits. In total, this circuit has one initial $|T\rangle$ state, 13T gates, and 15 two-qubit (CNOT or CZ) gates.

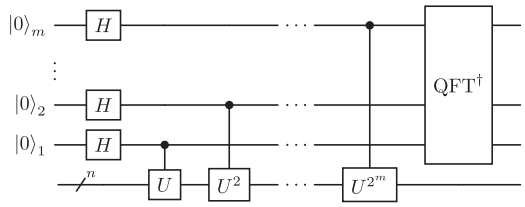
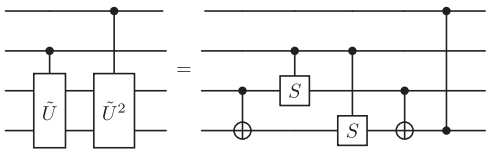


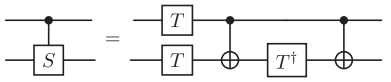
FIG. 19. The circuit for performing quantum phase estimation (QPE) to m bits of precision, used in step (2) of the HHL algorithm (Sec. III B). Here, $U = e^{i2\pi A}$.

which implies that $\tilde{U}^2 = CX(Z \otimes Z)CX = I \otimes Z$, and, in turn,



(D4)

Here, the controlled-S gates can be implemented using three T gates:



(D5)

2. Eigenvalue inversion

Here, we describe the eigenvalue inversion circuit that we use to implement the eigenvalue transformation in step (3) of the HHL algorithm, namely,

$$|\lambda_j\rangle \to |\lambda_j\rangle \otimes \left[\tilde{\lambda}_j^{-1}|0\rangle + \sqrt{1-\tilde{\lambda}_j^{-2}}|1\rangle\right], \tag{D6}$$

where j indexes an eigenvector (with eigenvalue λ_j) of the matrix A that is to be inverted by HHL and $\tilde{\lambda}_j^{-1} = C/\lambda$ for some constant C that enforces $|\tilde{\lambda}_j^{-1}| \leq 1$ (unless $\lambda_j = 0$, for which we define $\tilde{\lambda}_j^{-1} = 0$).

We consider the case in which eigenvalues $\lambda \in \{0, \frac{1}{4}, \frac{1}{2}\}$ are stored in a two-qubit register, $|\lambda_j\rangle = |a_j\rangle \otimes |b_j\rangle$, where we associate each value $\lambda_j \sim (a_j, b_j) \in \{0,1\}^2$ with a binary decimal encoding:

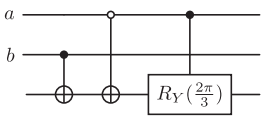
$$0 \sim (0,0), \qquad \frac{1}{4} \sim (0,1), \qquad \frac{1}{2} \sim (1,0), \qquad \frac{3}{4} \sim (1,1). \tag{D7}$$

Choosing $C = 1/4$, the desired transformations of a $|0\rangle$-state ancilla for eigenvalue inversion are

λ_j	(a_j, b_j)	$\tilde{\lambda}_j^{-1}$	final ancilla state
0	$(0,0)$	0	$\|1\rangle$
$\frac{1}{4}$	$(0,1)$	1	$\|0\rangle$
$\frac{1}{2}$	$(1,0)$	1/2	$\frac{1}{2}\|0\rangle + \frac{\sqrt{3}}{2}\|1\rangle$

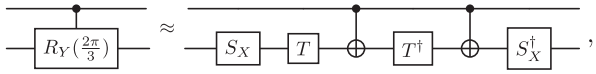
(D8)

where the target ancilla state for $\lambda_j = \frac{1}{2}$ is $R_Y[(2\pi)/3]|0\rangle$. These transformations are implemented coherently by the circuit



(D9)

Using the Solovay-Kitaev algorithm [104,105], we find that controlled-$R_Y[(2\pi)/3]$ gate can be approximated to a channel fidelity of approximately 0.966 by the Clifford + T decomposition



(D10)

where $S_X = R_X(\pi/2) = \mathrm{HSH}$.

APPENDIX E: THE RAMSEY PROTOCOL WITH A DEPHASING NOISE MODEL

Here, we describe a single-parameter dephasing model, which we call DepModel in the main text, of a Ramsey protocol in which repeated T gates are applied to an initial $|+\rangle$ state. For a fixed error parameter p, this model predicts that the Ramsey fidelity $\mathcal{F}_{T,p}(n)$ (that is, the fidelity of the noisily prepared state with respect to the noiseless target state) decays exponentially with respect to the number of applied T gates, n.

After Pauli twirling, a noisy T gate can be described by the single-qubit channel [89]

$$\mathcal{T}_p = (1-p)\mathcal{T} + p\mathcal{D}_z, \tag{E1}$$

where $\mathcal{T}$ is the channel for a noiseless T gate and $\mathcal{D}_z$ is a dephasing channel, which act on a single-qubit density matrix σ as

$$\begin{aligned} \mathcal{T}(\sigma) &= T\sigma T^\dagger, \\ \mathcal{D}_z(\sigma) &= \frac{1}{2}(\sigma + Z\sigma Z) = \sigma_{00}|0\rangle\langle 0| + \sigma_{11}|1\rangle\langle 1|. \end{aligned} \tag{E2}$$

Here, $\sigma_{ab} = \langle a|\sigma|b\rangle$, and the sum of channels A and B acts on a density matrix as $(A+B)(\rho) = A(\rho) + B(\rho)$. The parameter $p \in [0,1]$ can be interpreted as the probability of a dephasing event in which all relative phase information is

lost—in which case σ is replaced by $\mathcal{D}_z(\sigma)$—when applying a noisy T gate. Note that the definition of the noisy T gate in Eq. (E1) is slightly different from that in Eq. (5) in Ref. [26]; the definition used here is chosen in order to leverage properties of the dephasing channel $\mathcal{D}_z$ below.

The primary claim of this section is the following.

Lemma 1. If σ is the density matrix of a pure state that is an equal superposition of $|0\rangle$ and $|1\rangle$, for which $\sigma_{00} = \sigma_{11} = 1/2$, then the fidelity of the noisy state $\mathcal{T}_p^n(\sigma)$ with respect to the noiseless target state $\mathcal{T}^n(\sigma) = T^n \sigma T^{\dagger n}$ is

$$\mathcal{F}_{T,p}(n) = \frac{1}{2} + \frac{1}{2}(1-p)^n. \tag{E3}$$

Proof. Observe that

(a) the dephasing channel deletes phase information, meaning that $\mathcal{T} \circ \mathcal{D}_z = \mathcal{D}_z \circ \mathcal{T} = \mathcal{D}_z$;
(b) the dephasing channel is a projection, meaning that $\mathcal{D}_z^2 = \mathcal{D}_z$.

Together, these facts allow us to expand an n-fold application of the noisy T gate as

$$\mathcal{T}_p^n = (1-p)^n \mathcal{T}^n + [1-(1-p)^n]\mathcal{D}_z. \tag{E4}$$

If σ is a pure state, then $\mathcal{T}^n(\sigma)$ is a pure state, whose fidelity $\mathcal{F}_{T,p}(n)$ with the noisy state $\mathcal{T}_p^n(\sigma)$ is

$$\mathcal{F}_{T,p}(n) = \mathrm{Tr}[\mathcal{T}_p^n(\sigma)\mathcal{T}^n(\sigma)] \tag{E5}$$

$$= (1-p)^n \mathrm{Tr}[\mathcal{T}^n(\sigma)\mathcal{T}^n(\sigma)] + [1-(1-p)^n]\mathrm{Tr}[\mathcal{D}_z(\sigma)\mathcal{T}^n(\sigma)] \tag{E6}$$

$$= (1-p)^n + [1-(1-p)^n](\sigma_{00}^2 + \sigma_{11}^2). \tag{E7}$$

Substituting $\sigma_{00} = \sigma_{11} = 1/2$ yields $\mathcal{F}_{T,p}(n) = \frac{1}{2} + \frac{1}{2}(1-p)^n$. ∎

Note that the infidelity of a single noisy T gate acting on the initial state $|+\rangle\langle+|$ is $\epsilon = 1 - \mathcal{F}_{T,p}(1) = p/2$. Substituting $p = 2\epsilon$ into Eq. (E3) recovers the expression in Eq. (6) in Ref. [26].

APPENDIX F: NONEXPONENTIAL DECAY OF THE RAMSEY FIDELITY

The dephasing model in Appendix E assumes that logical T gates cause IID logical errors and, consequently, predicts an exponential decay of the Ramsey fidelity. This model agrees with simulations of Ramsey sequences in which every T gate is followed by a QEC cycle but fails to capture the nonexponential decay of Ramsey fidelity for sequences of T gates without QEC cycles (Fig. 10). Fundamentally, this nonexponential decay is due the fact that the logical T gates considered in this work are *transparent* to Z errors. That is, a physical Z error on a data qubit at the input to the logical T-gate gadget in Fig. 4(d) propagates to a physical Z error at the output of this gadget. A logical Z error can, therefore, be caused by pairs of physical Z errors that occur at distant T gates. Said otherwise, the entire Ramsey sequence can tolerate only one data-qubit Z error.

Consider a Ramsey sequence of n logical T gates, in which every logical T dephases each data qubit with IID probability g. By a derivation analogous to that of Eq. (E4) in Appendix E, the probability that a given data qubit has been dephased at the end of the sequence is

$$r = 1 - (1-g)^n \approx gn. \tag{F1}$$

Recalling that the dephasing channel is $D_z(\sigma) = \frac{1}{2}(\sigma + Z\sigma Z)$, a probability r of a dephasing event corresponds to a probability $\gamma = r/2$ of a Z error ($\sigma \to Z\sigma Z$). The probability Γ of an uncorrectable Z error at the end of a Ramsey sequence can be found by enumerating all configurations of uncorrectable multiqubit phase errors in the Steane code:

$$\Gamma = \frac{7}{2}\gamma^2(1-\gamma)^5 + 7\gamma^3(1-\gamma)^4 + (7\times 4)\gamma^4(1-\gamma)^3 + 7\gamma^6(1-\gamma) + \gamma^7 \approx 21\gamma^2 \approx \frac{21}{4}g^2n^2. \tag{F2}$$

In turn, the probability of a logical dephasing event at the end of the Ramsey sequence is $P = 2\Gamma$. Additionally accounting for an IID probability p of a logical dephasing event at each logical T gate, the overall Ramsey fidelity becomes

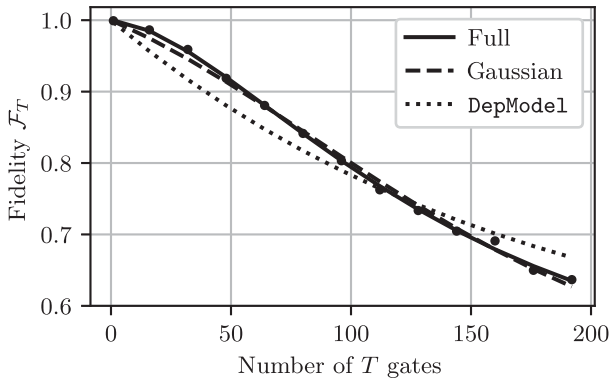


FIG. 20. Fidelity of Ramsey sequences without QEC cycles and fits to the single-parameter DepModel in Eq. (E3) (dotted line), the two-parameter Gaussian approximation in Eq. (F3) (dashed line), and the full two-parameter model in Eq. (F3) (solid line). The simulation data here are the same as the "without QEC" data in Fig. 10.

$$\mathcal{F}_{\text{Ramsey}}(p,g,n) = \frac{1}{2} + \frac{1}{2}(1-P)(1-p)^n \approx \frac{1}{2} + \frac{1}{2}\exp\left(-pn - \frac{21}{2}g^2n^2\right). \quad \text{(F3)}$$

Figure 20 shows the fidelity of Ramsey sequences without QEC cycles (same as the "without QEC" data in Fig. 10 in the main text), together with fits to both the full expression (in terms of $P = 2\Gamma$) and the Gaussian approximation in Eq. (F3). Notably, the full expression in Eq. (F3), written as a function of (g, p, n) without taking any of the approximations in Eqs. (F1)–(F3), provides a markedly better fit to early-time behavior than the Gaussian approximation in Eq. (F3). We use the full expression in Eq. (F3) for the fit in Fig. 10 in the main text.

APPENDIX G: ENTANGLEMENT FIDELITY OF A SINGLE-QUBIT CHANNEL

The entanglement fidelity of a single-qubit channel $\mathcal{C}$ is [93]

$$\mathcal{F}(\mathcal{C}) = \langle\phi|(\mathcal{I}\otimes\mathcal{C})(\phi)|\phi\rangle = \mathrm{Tr}[(\mathcal{I}\otimes\mathcal{C})(\phi)\phi], \quad \text{(G1)}$$

where $|\phi\rangle = (1/\sqrt{2})\sum_{a\in\{0,1\}}|aa\rangle$ is a maximally entangled state of two qubits and $\phi = |\phi\rangle\langle\phi|$. Unlike the commonly used gate fidelity, the entanglement fidelity penalizes the deleterious effect of gate errors on the entanglement between the target qubits of a gate and other qubits in a quantum computation [93]. For reference, the entanglement fidelity $\mathcal{F}_{\mathcal{G}}$ of a noisy channel $\mathcal{G}$ acting on a d-dimensional system ($d = 2$ for a qubit) is related to the gate fidelity $\mathcal{F}_{\mathcal{G}}^{\text{gate}}$ by $\mathcal{F}_{\mathcal{G}}^{\text{gate}} = (d\mathcal{F}_{\mathcal{G}} + 1)/(d+1)$ [106,107].

Lemma 2. The entanglement fidelity of the single-qubit channel $\mathcal{C}$ can be written as

$$\mathcal{F}(\mathcal{C}) = \frac{1}{4}\sum_{\substack{P\in\{X,Y,Z\}\\ b\in\{0,1\}}}\mathrm{Tr}[\mathcal{C}(\Pi_{P,b})\Pi_{P,b}] - \frac{1}{2}, \quad \text{(G2)}$$

where $\Pi_{P,b} = [I + (-1)^b P]/2$ is the projector into the eigenstate of the Pauli operator P with eigenvalue $(-1)^b$, for which $P = \Pi_{P,0} - \Pi_{P,1}$.

Proof. By direct substitution,

$$\mathcal{F}(\mathcal{C}) = \frac{1}{4}\sum_{a,b,c,d\in\{0,1\}}\mathrm{Tr}\Big[(|a\rangle\langle b|\otimes\mathcal{C}(|a\rangle\langle b|)) \times (|c\rangle\langle d|\otimes|c\rangle\langle d|)\Big] \quad \text{(G3)}$$

$$= \frac{1}{4}\sum_{a,b,c,d\in\{0,1\}}\mathrm{Tr}\Big[|a\rangle\langle b||c\rangle\langle d|\times\mathrm{Tr}[\mathcal{C}(|a\rangle\langle b||c\rangle\langle d|)\Big], \quad \text{(G4)}$$

where we used the fact that $\mathrm{Tr}(A\otimes B) = \mathrm{Tr}(A)\times\mathrm{Tr}(B)$. Here, $\mathrm{Tr}[|a\rangle\langle b||c\rangle\langle d|] = 1$ if $(c,d) = (b,a)$ and 0 otherwise, so

$$\mathcal{F}(\mathcal{C}) = \frac{1}{4}\sum_{a,b\in\{0,1\}}\mathrm{Tr}[\mathcal{C}(|a\rangle\langle b||b\rangle\langle a|]. \quad \text{(G5)}$$

The sum over $|a\rangle\langle b|$ can be replaced by a sum over any orthonormal basis $\mathcal{B}_2$ for the space of linear operators on a single-qubit Hilbert space, which is to say that

$$\mathcal{F}(\mathcal{C}) = \frac{1}{4}\sum_{M\in\mathcal{B}_2}\mathrm{Tr}[\mathcal{C}(M)M^\dagger], \quad \text{(G6)}$$

where $\mathcal{B}_2$ is any basis of 2×2 matrices for which $A, B\in\mathcal{B}_2$ implies $\mathrm{Tr}(AB^\dagger) = 1$ iff $A = B$ and 0 otherwise. Considering, in particular, the Pauli basis $\{I, X, Y, Z\}/\sqrt{2}$, we use the fact that $\mathcal{C}$ is a trace-preserving map, meaning $\mathrm{Tr}[\mathcal{C}(I)] = 2$, and expand $P = \Pi_{P,0} - \Pi_{P,1} = \sum_{b\in\{0,1\}}(-1)^b\Pi_{P,b}$ for all $P\in\{X,Y,Z\}$ to write

$$\mathcal{F}(\mathcal{C}) = \frac{1}{8}\mathrm{Tr}[\mathcal{C}(I)] + \frac{1}{8}\sum_{P\in\{X,Y,Z\}}\mathrm{Tr}[\mathcal{C}(P)P] \quad \text{(G7)}$$

$$= \frac{1}{4} + \frac{1}{8}\sum_{\substack{P\in\{X,Y,Z\}\\ b\in\{0,1\}}}(-1)^b\mathrm{Tr}[\mathcal{C}(\Pi_{P,b})P]. \quad \text{(G8)}$$

Substituting $(-1)^b P = 2\Pi_{P,b} - I$, we then find that

$$\mathcal{F}(\mathcal{C}) = \frac{1}{4} + \frac{1}{4}\sum_{\substack{P\in\{X,Y,Z\}\\ b\in\{0,1\}}}\mathrm{Tr}[\mathcal{C}(\Pi_{P,b})\Pi_{P,b}] - \frac{1}{8}\sum_{\substack{P\in\{X,Y,Z\}\\ b\in\{0,1\}}}\mathrm{Tr}[\mathcal{C}(\Pi_{P,b})], \quad \text{(G9)}$$

where $\mathrm{Tr}[\mathcal{C}(\Pi_{P,b})] = \mathrm{Tr}(\Pi_{P,b}) = 1$ because $\mathcal{C}$ is trace preserving, so the final sum $\sum_{P,b}\mathrm{Tr}[\mathcal{C}(\Pi_{P,b})] = 6$, and altogether

$$\mathcal{F}(\mathcal{C}) = \frac{1}{4}\sum_{\substack{P\in\{X,Y,Z\}\\ b\in\{0,1\}}}\mathrm{Tr}[\mathcal{C}(\Pi_{P,b})\Pi_{P,b}] - \frac{1}{2}. \quad \text{(G10)}$$

■

Lemma 3. If $\mathcal{Q}_p = (1-p)\mathcal{I} + p\mathcal{D}$, where $p\in[0,1]$ is a scalar and $\mathcal{D}$ is a single-qubit channel that maps any input states to the maximally mixed state $I/2$, then

$$\mathcal{F}(\mathcal{Q}_p^n) = \frac{1}{4} + \frac{3}{4}(1-p)^n. \quad \text{(G11)}$$

Proof. Let $\mathcal{Q} = (1-p)\mathcal{I} + p\mathcal{D}$. Similarly to the proof of Lemma 1, observe that $\mathcal{I}\circ\mathcal{D} = \mathcal{D}\circ\mathcal{I} = \mathcal{D}^2 = \mathcal{D}$, so

$$\mathcal{Q}_p^n = (1-p)^n\mathcal{I} + [1-(1-p)^n]\mathcal{D}, \tag{G12}$$

which implies that

$$\begin{aligned}\mathrm{Tr}[\mathcal{Q}_p^n(\Pi_{P,b})\Pi_{P,b}] &= (1-p)^n \times \mathrm{Tr}[\Pi_{P,b}^2] \\ &\quad + [1-(1-p)^n] \times \frac{1}{2}\mathrm{Tr}[\Pi_{P,b}] \\ &= \frac{1}{2} + \frac{1}{2}(1-p)^n. \end{aligned} \tag{G13}$$

By direct substitution into Eq. (G10) (Lemma 2),

$$\mathcal{F}(\mathcal{Q}_p^n) = \frac{1}{4} + \frac{3}{4}(1-p)^n. \tag{G14}$$

■

APPENDIX H: ADDITIONAL BENCHMARKING DATA

Tables I–V provide detailed emulator and hardware data collected for the application benchmarks presented in Sec. IV in the main text. Data tables indicate the number of logical qubits in an algorithm (N), the number of T gates in the executed circuit after decomposition to a Clifford + T gate set (n_T), and, if applicable, the QAOA depth p. For reference, data tables also provide observables of interest for noise-free (NF) circuits both before (original) and after (decomposed) decomposition to a Clifford + T gate set. Results for unencoded (physical, decomposed) QAOA circuits have "$\cdots$" for their RUS limit and number of QEC cycles, while all other circuits are encoded into the Steane code. QEC cycles are performed using the Steane-swap gadget in Fig. 4(d), for which logical $|0\rangle$ and $|+\rangle = H|0\rangle$ states are prepared using the circuit in Fig. 2. Logical T gates are implemented using the T-swap gadget in Fig. 4(b), for which logical $|T\rangle = HS|H\rangle$ states are prepared using the circuit in Fig. 16. "Total shots" refers to the total number of shots executed on hardware (or an emulator), while "kept shots" refers to the number of shots in which all state-preparation subroutines succeeded (indicated by zero measurement outcomes on the appropriate flag or ancilla qubits).

The QAOA data tables also report the LABS *merit factor* of a bit string $s \in \{+1,-1\}^N$, which is defined as

$$F_{\mathrm{LABS}}(s) = \frac{N^2}{2E_{\mathrm{LABS}}(s)}, \tag{H1}$$

and is conjectured to approach $F_{\mathrm{LABS}}^{\mathrm{opt}} \approx 12.32$ as $N \to \infty$ [108]. The merit factor of a QAOA state $|\boldsymbol{\beta},\boldsymbol{\gamma}\rangle$ is

$$F_{\mathrm{LABS}}(\boldsymbol{\beta},\boldsymbol{\gamma}) = \frac{N^2}{2\langle\boldsymbol{\beta},\boldsymbol{\gamma}|H_{\mathrm{LABS}}|\boldsymbol{\beta},\boldsymbol{\gamma}\rangle}. \tag{H2}$$

TABLE I. Results from running QAOA for $N = 5$ LABS on H2-2 and Helios computers as well as the H2-2E emulator. Circuits with QEC cycles apply one QEC cycle to each logical qubit in the middle of the logical circuit, evenly partitioning the logical two-qubit gates.

Logical qubits	$N=5$											
Circuit parameters	$(p,n_T)=(1,9)$		$(p,n_T)=(2,5)$				$(p,n_T)=(2,9)$					
Success probability (original, NF)	0.782		0.935				0.935					
Success probability (decomposed, NF)	0.694		0.625				0.854					
Merit factor (original, NF)	5.174		5.931				5.931					
Merit factor (decomposed, NF)	4.641		4.375				5.396					
Device	H2-2E	H2-2	H2-2				H2-2				Helios	
RUS limit	$\cdots$	1	$\cdots$	1	$\cdots$	1	1	2	1	1	2	
QEC cycles	$\cdots$	0	$\cdots$	0	$\cdots$	0	5	0	0	5	0	
Physical qubits	5	48	5	48	5		48			48		
Physical two-qubit gates	11	474	22	399	22	551	731	≤ 885	551	731	≤ 885	
Estimated NF run-time (seconds)									1.37	1.92		
Success probability	0.522(22)	0.466(20)	0.484(22)	0.453(29)	0.592(22)	0.587(19)	0.491(31)	0.500(23)	0.545(19)	0.567(19)	0.544(22)	
Merit factor	4.65(11)	4.25(10)	4.43(11)	4.15(15)	5.34(09)	4.97(09)	4.39(16)	4.53(11)	4.83(10)	4.83(10)	4.74(11)	
Kept/total shots	500/500	625/1000	500/500	285/400	500/500	664/1000	267/500	488/500	662/1000	652/1000	493/500	
Postselection rate	1	0.625	1	0.7125	1	0.664	0.534	0.976	0.662	0.652	0.986	

TABLE II. Results from running QAOA for $N = 6$ LABS on H2-2 and Helios computers as well as the H2-2E emulator. Circuits with QEC cycles apply one QEC cycle to each logical qubit in the middle of the logical circuit, evenly partitioning the logical two-qubit gates.

Logical qubits	$N = 6$									
Circuit parameters	$(p, n_T) = (1, 13)$				$(p, n_T) = (1, 6)$			$(p, n_T) = (2, 0)$		
Success probability (original, NF)	0.915				0.915			1		
Success probability (decomposed, NF)	0.979				0.672			1		
Merit factor (original, NF)	2.438				2.438			2.571		
Merit factor (decomposed, NF)	2.523				1.998			2.571		
Device	H2-2		Helios		H2-2E	H2-2	H2-2	Helios		
RUS limit	⋯	1	1	2	⋯	1	1	⋯	1	1
QEC cycles	⋯	0	0	0	⋯	0	0	⋯	0	6
Physical qubits	6	55	55		6	55	55	6	55	
Physical two-qubit gates	18	686	686	≤ 1155	18	420	318	36	318	618
Estimated NF run-time (seconds)			1.78					0.19	0.64	1.33
Success probability	0.957(12)	0.784(28)	0.562(33)	0.628(22)	0.668(21)	0.582(27)	0.893(15)	0.982(06)	0.935(11)	0.860(17)
Merit factor	2.484(24)	2.252(42)	1.927(50)	2.017(34)	1.985(38)	1.874(47)	2.417(22)	2.546(08)	2.481(16)	2.378(23)
Kept/total shots	300/300	218/500	226/500	479/500	500/500	323/500	422/500	500/500	474/500	435/500
Postselection rate	1	0.436	0.452	0.958	1	0.646	0.844	1	0.948	0.870

TABLE III. Results from running QAOA for $N = 7, 8, 9$ LABS on the Helios computer.

Logical qubits	$N = 7$		$N = 8$		$N = 9$	
Circuit parameters	$(p, n_T) = (1, 0)$		$(p, n_T) = (1, 8)$		$(p, n_T) = (1, 25)$	
Success probability (original, NF)	0.118		0.170		0.122	
Success probability (decomposed, NF)	0.125		0.125		0.108	
Merit factor (original, NF)	2.792		1.806		1.862	
Merit factor (decomposed, NF)	2.880		1.728		1.764	
Device	Helios					
RUS limit	⋯	1	⋯	1	⋯	1
QEC cycles	⋯	0	⋯	0	⋯	0
Physical qubits	7	62	8	69	9	76
Physical two-qubit gates	37	336	49	735	78	1595
Estimated NF run-time (seconds)	0.17	0.69	0.21	1.69	0.29	3.78
Success probability	0.136(15)	0.109(14)	0.102(14)	0.108(17)	0.130(15)	0.061(20)
Merit factor	2.998(101)	2.770(97)	1.710(47)	1.724(58)	1.842(40)	1.682(20)
Kept/total shots	500/500	469/500	500/500	332/500	500/500	148/500
Postselection rate	1	0.938	1	0.664	1	0.296

TABLE IV. Results from running QAOA on a selected $N = 12$ portfolio optimization instance on the Helios computer.

Logical qubits	$N = 12$			
Circuit parameters	$(p, n_T) = (1, 2)$		$(p, n_T) = (2, 4)$	
Success probability (original, NF)	0.2672		0.7077	
Success probability (decomposed, NF)	0.1821		0.2124	
Success probability (random guess)	$1/2^{12} \approx 0.00025$			
Approximation ratio (original, NF)	0.9258		0.9844	
Approximation ratio (decomposed, NF)	0.9481		0.9569	
Approximation ratio (random guess)	0.5038			
Device	Helios			
RUS limit	⋯	1	⋯	1
QEC cycles	⋯	0	⋯	0
Physical qubits	12	97	12	97
Physical two-qubit gates	132	1132	264	2132
Estimated NF run-time (seconds)	0.267	1.64	0.436	2.66
Success probability	0.136(5)	0.131(17)	0.180(7)	0.034(10)
Approximation ratio	0.9224(45)	0.8743(64)	0.9093(60)	0.6991(115)
Kept/total shots	500/500	411/500	500/500	349/500
Postselection rate	1	0.822	1	0.698

TABLE V. Results from running the decomposed HHL circuit in Fig. 18 on the Helios computer. Data with an effective RUS limit of r_{eff} is postselected on the shots in which all RUS routines succeeded in at most r_{eff} attempts, thereby controlling for the effects of a different RUS limit in the program submitted to hardware. Uncertainties in the final digits of fidelities reflect 68% confidence intervals determined by bootstrapping. The total number of shots for all experiments is 1000. In addition to the number of kept shots (those in which all state-preparation subroutines succeeded, indicated by zero measurement outcomes on the appropriate flag or ancilla qubits, SP = 0), more granular data are provided on the number of shots that were kept and have logical QPE and EV measurement outcomes as indicated. Fidelities are computed from the shots with SP = QPE = EV = 0.

Logical qubits	$N = 5$														
Circuit parameters	$n_T = 14$														
Device	Helios														
RUS limit	…		1			2				3					
Effective RUS limit	…		1		1		2		1		3		1	2	3
QEC cycles	…	0	2	4		0				0				2	
Physical qubits	5		48			48				48				48	
Physical two-qubit gates	15	677	749	821	677		≤ 1166	677		≤ 1166	≤ 1655		749	≤ 1282	≤ 1815
Estimated NF run-time (seconds)	0.09	1.79	2.02	2.26	1.83			1.83					2.06		
Fidelity $\mathcal{F}_{\|x\rangle}$	0.9995(115)	0.901(30)	0.966(23)	0.871(38)	0.825(36)	0.832(26)		0.778(40)	0.778(40)		0.773(27)		0.839(38)	0.830(28)	0.827(27)
Kept shots (SP = 0)	1000	500	511	412	505	970		470	960		998		488	967	999
Kept shots w/ QPE = 0, EV = 0	748	272	293	223	240	441		226	449		466		214	401	415
Kept shots w/ QPE = 0, EV = 1	238	88	110	89	104	209		89	198		205		89	188	194
Kept shots w/ QPE ≠ 0, EV = 0	5	93	65	64	93	185		93	174		184		99	213	222
Kept shots w/ QPE ≠ 0, EV = 1	9	47	43	36	68	135		62	139		143		86	165	168

[1] R. Acharya, D. A. Abanin, L. Aghababaie-Beni *et al.*, *Quantum error correction below the surface code threshold*, Nature (London) **638**, 920 (2025).

[2] T. He, W. Lin, R. Wang *et al.*, *Experimental quantum error correction below the surface code threshold via all-microwave leakage suppression*, Phys. Rev. Lett. **135**, 260601 (2025).

[3] B. W. Reichardt, D. Aasen, R. Chao *et al.*, *Demonstration of quantum computation and error correction with a tesseract code*, arXiv:2409.04628.

[4] A. Paetznick, M. P. da Silva, C. Ryan-Anderson *et al.*, *Demonstration of logical qubits and repeated error correction with better-than-physical error rates*, arXiv:2404.02280.

[5] J. M. Pino, J. M. Dreiling, C. Figgatt *et al.*, *Demonstration of the trapped-ion quantum CCD computer architecture*, Nature (London) **592**, 209 (2021).

[6] S. A. Moses, C. H. Baldwin, M. S. Allman *et al.*, *A race-track trapped-ion quantum processor*, Phys. Rev. X **13**, 041052 (2023).

[7] A. Ransford, M. S. Allman, J. Arkinstall *et al.*, *Helios: A 98-qubit trapped-ion quantum computer*, arXiv:2511.05465.

[8] D. Bluvstein, S. J. Evered, A. A. Geim *et al.*, *Logical quantum processor based on reconfigurable atom arrays*, Nature (London) **626**, 58 (2024).

[9] B. W. Reichardt, A. Paetznick, D. Aasen *et al.*, *Fault-tolerant quantum computation with a neutral atom processor*, arXiv:2411.11822.

[10] A. Radnaev, W. Chung, D. Cole *et al.*, *Universal neutral-atom quantum computer with individual optical addressing and nondestructive readout*, PRX Quantum **6**, 030334 (2025).

[11] D. Bluvstein, A. A. Geim, S. H. Li *et al.*, *A fault-tolerant neutral-atom architecture for universal quantum computation*, Nature (London) **649**, 39 (2025).

[12] S. Bravyi, D. Poulin, and B. Terhal, *Tradeoffs for reliable quantum information storage in 2D systems*, Phys. Rev. Lett. **104**, 050503 (2010).

[13] N. Baspin and A. Krishna, *Quantifying nonlocality: How outperforming local quantum codes is expensive*, Phys. Rev. Lett. **129**, 050505 (2022).

[14] S. Dai and R. Li, *Locality vs quantum codes*, in *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25 (Association for Computing Machinery, New York, USA, 2025), pp. 677–688.

[15] H. Sayginel, S. Koutsioumpas, M. Webster, A. Rajput, and D. E. Browne, *Fault-tolerant logical Clifford gates from code automorphisms*, PRX Quantum **6**, 030343 (2025).

[16] M. Cain, C. Zhao, H. Zhou, N. Meister, J. P. Ataides, A. Jaffe, D. Bluvstein, and M. D. Lukin, *Correlated decoding of logical algorithms with transversal gates*, Phys. Rev. Lett. **133**, 240602 (2024).

[17] H. Zhou, C. Zhao, M. Cain *et al.*, *Low-overhead transversal fault tolerance for universal quantum computation*, Nature (London) **646**, 303 (2025).

[18] M. Cain, D. Bluvstein, C. Zhao *et al.*, *Fast correlated decoding of transversal logical algorithms*, arXiv:2505.13587.
[19] J. P. B. Ataides, A. Gu, S. F. Yelin, and M. D. Lukin, *Neural decoders for universal quantum algorithms*, PRX Intell. **1**, 013008 (2026).
[20] A. Steane, *Multiple-particle interference and quantum error correction*, Proc. R. Soc. A **452**, 2551 (1996).
[21] B. W. Reichardt, *Fault-tolerant quantum error correction for Steane's seven-qubit color code with few or no extra qubits*, Quantum Sci. Technol. **6**, 015007 (2020).
[22] H. Goto, *Minimizing resource overheads for fault-tolerant preparation of encoded states of the Steane code*, Sci. Rep. **6**, 19578 (2016).
[23] C. Ryan-Anderson, J. G. Bohnet, K. Lee *et al.*, *Realization of real-time fault-tolerant quantum error correction*, Phys. Rev. X **11**, 041058 (2021).
[24] L. Postler and S. Heuβen, I. Pogorelov *et al.*, *Demonstration of fault-tolerant universal quantum gate operations*, Nature (London) **605**, 675 (2022).
[25] S. Dasu, S. Burton, K. Mayer *et al.*, *Breaking even with magic: Demonstration of a high-fidelity logical non-Clifford gate*, arXiv:2506.14688.
[26] K. Mayer, C. Ryan-Anderson, N. Brown *et al.*, *Benchmarking logical three-qubit quantum Fourier transform encoded in the Steane code on a trapped-ion quantum computer*, arXiv:2404.08616.
[27] K. Yamamoto, Y. Kikuchi, D. Amaro *et al.*, *Quantum error-corrected computation of molecular energies*, PRX Quantum **7**, 020319 (2026).
[28] W. Dai and D. Berleant, *Benchmarking contemporary deep learning hardware and frameworks:A survey of qualitative metrics*, in *2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI)* (2019), pp. 148–155, arXiv:1907.03626.
[29] M. H. Abobeih, Y. Wang, J. Randall *et al.*, *Fault-tolerant operation of a logical qubit in a diamond quantum processor*, Nature (London) **606**, 884 (2022).
[30] Y. Wang, S. Simsek, T. M. Gatterman *et al.*, *Fault-tolerant one-bit addition with the smallest interesting color code*, Sci. Adv. **10**, eado9024 (2024).
[31] N. Lacroix, A. Bourassa, F. J. H. Heras *et al.*, *Scaling and logic in the color code on a superconducting quantum processor*, Nature (London) **645**, 614 (2025).
[32] S. Dasu, M. DeCross, A. Y. Guo *et al.*, *Computing with many encoded logical qubits beyond break-even*, arXiv:2602.22211.
[33] K. Yamamoto, S. Duffield, Y. Kikuchi, and D. Munoz Ramo, *Demonstrating Bayesian quantum phase estimation with quantum error detection*, Phys. Rev. Res. **6**, 013221 (2024).
[34] D. Hangleiter, M. Kalinowski, D. Bluvstein, M. Cain, N. Maskara, X. Gao, A. Kubica, M. D. Lukin, and M. J. Gullans, *Fault-tolerant compiling of classically hard IQP circuits on hypercubes*, PRX Quantum **6**, 020338 (2025).
[35] E. Farhi, J. Goldstone, and S. Gutmann, *A quantum approximate optimization algorithm*, arXiv:1411.4028.
[36] K. Blekos, D. Brand, A. Ceschini *et al.*, *A review on quantum approximate optimization algorithm and its variants*, Phys. Rep. **1068**, 1 (2024).
[37] A. Boehmer, *Binary pulse compression codes*, IEEE Trans. Inf. Theory **13**, 156 (1967).
[38] M. Schroeder, *Synthesis of low-peak-factor signals and binary sequences with low autocorrelation (Corresp.)*, IEEE Trans. Inf. Theory **16**, 85 (1970).
[39] A. W. Harrow, A. Hassidim, and S. Lloyd, *Quantum algorithm for linear systems of equations*, Phys. Rev. Lett. **103**, 150502 (2009).
[40] P. Aliferis, D. Gottesman, and J. Preskill, *Quantum accuracy threshold for concatenated distance-3 codes*, arXiv:quant-ph/0504218.
[41] D. Gottesman, *An introduction to quantum error correction and fault-tolerant quantum computation*, arXiv:0904.2557.
[42] A. R. Calderbank and P. W. Shor, *Good quantum error-correcting codes exist*, Phys. Rev. A **54**, 1098 (1996).
[43] H. Bombin, *An introduction to topological quantum codes*, arXiv:1311.0277.
[44] A. Kubica and M. E. Beverland, *Universal transversal gates with color codes: A simplified approach*, Phys. Rev. A **91**, 032330 (2015).
[45] C. Chamberland and A. W. Cross, *Fault-tolerant magic state preparation with flag qubits*, Quantum **3**, 143 (2019).
[46] S. Heußen, L. Postler, M. Rispler, I. Pogorelov, C. D. Marciniak, T. Monz, P. Schindler, and M. Muller, *Strategies for a practical advantage of fault-tolerant circuit design in noisy trapped-ion quantum computers*, Phys. Rev. A **107**, 042422 (2023).
[47] Y. Ibe, Y. Hirano, Y. Ozu *et al.*, *Measurement-based fault-tolerant quantum computation on high-connectivity devices: A resource-efficient approach toward Early FTQC*, arXiv:2510.18652.
[48] C. Ryan-Anderson, Pecos, https://github.com/PECOS-packages/PECOS (2018).
[49] R. Zen, J. Olle, L. Colmenarez, M. Puviani, M. Muller, and F. Marquardt, *Quantum circuit discovery for fault-tolerant logical state preparation with reinforcement learning*, Phys. Rev. X **15**, 041012 (2025).
[50] C. Chamberland and M. E. Beverland, *Flag fault-tolerant error correction with arbitrary distance codes*, Quantum **2**, 53 (2018).
[51] R. Chao and B. W. Reichardt, *Flag fault-tolerant error correction for any stabilizer code*, PRX Quantum **1**, 010302 (2020).
[52] S. Boulebnane and A. Montanaro, *Solving boolean satisfiability problems with the quantum approximate optimization algorithm*, PRX Quantum **5**, 030348 (2024).
[53] R. Shaydulin, C. Li, S. Chakrabarti *et al.*, *Evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem*, Sci. Adv. **10**, eadm6761 (2024).
[54] S. Boulebnane, A. Khan, M. Liu *et al.*, *Evidence that the quantum approximate optimization algorithm optimizes the Sherrington-Kirkpatrick model efficiently in the average case*, Phys. Rev. Lett. **136**, 240601 (2026).
[55] A. Apte, S. Boulebnane, Y. Jin *et al.*, *Quantum approximate optimization of integer graph problems and surpassing semidefinite programming for Max-k-Cut*, arXiv:2602.05956.

[56] E. Farhi, S. Gutmann, D. Ranard, and B. Villalonga, *Lower bounding the MaxCut of high girth 3-regular graphs using the QAOA*, arXiv:2503.12789.
[57] A. Montanaro and L. Zhou, *Quantum speedups in solving near-symmetric optimization problems by low-depth QAOA*, arXiv:2411.04979.
[58] S. H. Sureshbabu, D. Herman, R. Shaydulin *et al.*, *Parameter setting in quantum approximate optimization of weighted problems*, Quantum **8**, 1231 (2024).
[59] E. Farhi, J. Goldstone, S. Gutmann, and L. Zhou, *The quantum approximate optimization algorithm and the Sherrington-Kirkpatrick model at infinite size*, Quantum **6**, 759 (2022).
[60] J. Wurtz and D. Lykov, *Fixed-angle conjectures for the quantum approximate optimization algorithm on regular MaxCut graphs*, Phys. Rev. A **104**, 052419 (2021).
[61] Z. He, R. Shaydulin, D. Herman *et al.*, *Parameter Setting Heuristics Make the Quantum Approximate Optimization Algorithm Suitable for the Early Fault-Tolerant Era*, in *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, ICCAD '24 (Association for Computing Machinery, New York, 2025), pp. 1–7.
[62] Z. He, R. Raymond, R. Shaydulin, and M. Pistoia, *Nonvariational quantum random access optimization with alternating operator ansatz*, Sci. Rep. **15**, 29191 (2025).
[63] M. Golay, *Sieves for low autocorrelation binary sequences*, IEEE Trans. Inf. Theory **23**, 43 (1977).
[64] Z. Zhang, J. Shen, N. Kumar, and M. Pistoia, *New improvements in solving large LABS instances using massively parallelizable memetic Tabu search*, arXiv:2504.00987.
[65] B. Bošković, F. Brglez, and J. Brest, *Low-autocorrelation binary sequences: On improved merit factors and runtime predictions to achieve them*, Appl. Soft Comput. **56**, 262 (2017).
[66] B. Bošković, F. Brglez, and J. Brest, *A GitHub archive for solvers and solutions of the `labs` problem*, https://github.com/borkob/git_labs (2016).
[67] T. Packebusch and S. Mertens, *Low autocorrelation binary sequences*, J. Phys. A **49**, 165001 (2016).
[68] Qokit, https://github.com/jpmorganchase/QOKit (2025).
[69] A. M. Childs, R. Kothari, and R. D. Somma, *Quantum algorithm for systems of linear equations with exponentially improved dependence on precision*, SIAM J. Comput. **46**, 1920 (2017).
[70] S. Chakraborty, A. Gilyén, and S. Jeffery, *The power of block-encoded matrix powers: Improved regression techniques via faster Hamiltonian simulation*, *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)* (2019), 132, 33:1, arXiv:1804.01973.
[71] D. An and L. Lin, *Quantum linear system solver based on time-optimal adiabatic quantum computing and quantum approximate optimization algorithm*, ACM Trans. Quantum Comput. **3**, 5:1 (2022).
[72] Pedro C. Costa, D. An, Y. R. Sanders, Y. Su, R. Babbush, and D. W. Berry, *Optimal scaling quantum linear-systems solver via discrete adiabatic theorem*, PRX Quantum **3**, 040303 (2022).
[73] A. Montanaro and S. Pallister, *Quantum algorithms and the finite element method*, Phys. Rev. A **93**, 032324 (2016).
[74] J.-P. Liu, H. Ø. Kolden, H. K. Krovi *et al.*, *Efficient quantum algorithm for dissipative nonlinear differential equations*, Proc. Natl. Acad. Sci. U.S.A. **118**, e2026805118 (2021).
[75] N. Linden, A. Montanaro, and C. Shao, *Quantum vs. classical algorithms for solving the heat equation*, Commun. Math. Phys. **395**, 601 (2022).
[76] I. Kerenidis and A. Prakash, *Quantum recommendation systems*, arXiv:1603.08675.
[77] R. Babbush, D. W. Berry, R. Kothari, R. D. Somma, and N. Wiebe, *Exponential quantum speedup in simulating coupled classical oscillators*, Phys. Rev. X **13**, 041041 (2023).
[78] S. Bravyi, R. Manson-Sawko, M. Zayats, and S. Zhuk, *Quantum simulation of a noisy classical nonlinear dynamics*, arXiv:2507.06198.
[79] A. Y. Kitaev, *Quantum measurements and the Abelian stabilizer problem*, arXiv:quant-ph/9511026.
[80] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, Cambridge, England, 2010).
[81] T. Häner, M. Roetteler, and K. M. Svore, *Optimizing quantum circuits for arithmetic*, arXiv:1805.12445.
[82] B. Eastin and E. Knill, *Restrictions on transversal encoded quantum gate sets*, Phys. Rev. Lett. **102**, 110502 (2009).
[83] T. Hao, A. Xu, and S. Tannu, *Reducing T gates with unitary synthesis*, arXiv:2503.15843.
[84] S. Sivarajah, S. Dilkes, A. Cowtan *et al.*, *T|ket⟩: A retargetable compiler for NISQ devices*, Quantum Sci. Technol. **6**, 014003 (2020).
[85] J. Paykin, A. T. Schmitz, M. Ibrahim *et al.*, *PCOAST: A Pauli-Based Quantum Circuit Optimization Framework*, in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)* (2023), Vol. 01, pp. 715–726.
[86] A. T. Schmitz, Nicolas P. D. Sawaya, S. Johri, and A. Y. Matsuura, *Graph optimization perspective for low-depth Trotter-Suzuki decomposition*, Phys. Rev. A **109**, 042418 (2024).
[87] *System model h2 emulators—quantinuum systems*, https://docs.quantinuum.com/systems/user_guide/emulator_user_guide/emulators/h2_emulators.html (2025).
[88] J. A. Smolin, J. M. Gambetta, and G. Smith, *Efficient method for computing the maximum-likelihood quantum state from measurements with additive Gaussian noise*, Phys. Rev. Lett. **108**, 070502 (2012).
[89] 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).
[90] A. Ceasura, P. Iyer, J. J. Wallman, and H. Pashayan, *Non-exponential behaviour in logical randomized benchmarking*, arXiv:2212.05488.
[91] M. Koch, A. Lawrence, K. Singhal *et al.*, *GUPPY: Pythonic quantum-classical programming*, in *Informal Proceedings of the Fourth International Workshop on Programming Languages for Quantum Computing (PLanQC '24)* (2024), arXiv:2510.12582.

[92] M. Koch, A. Borgna, C. Roy *et al.*, *Imperative quantum programming with ownership and borrowing in Guppy*, in *Informal Proceedings of the Fifth International Workshop on Programming Languages for Quantum Computing (PLanQC '25)* (2025), arXiv:2510.13082.

[93] M. A. Nielsen, *The entanglement fidelity and quantum error correction*, arXiv:quant-ph/9606012.

[94] T. Hao, J. Sullivan, S. Omanakuttan *et al.*, *Compilation pipeline for predicting algorithmic break-even in an early-fault-tolerant surface code architecture*, arXiv:2511.20947.

[95] *ONISQ: Optimization with Noisy Intermediate-Scale Quantum devices*, https://www.darpa.mil/research/programs/optimization-with-noisy-intermediate-scale-quantum-devices (2019).

[96] R. Shaydulin and M. Pistoia, *Qaoa with* $n \cdot p \geq 200$, in *Proceedings of the 2023 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, New York, 2023), Vol. 01, pp. 1074–1077.

[97] E. Pelofske, A. Bärtschi, L. Cincio *et al.*, *Scaling whole-chip QAOA for higher-order ising spin glass models on heavy-hex graphs*, npj Quantum Inf. **10**, 109 (2024).

[98] E. Pelofske, A. Bärtschi, J. Golden, and S. Eidenbenz, *High-Round QAOA for MAX k-SAT on Trapped Ion NISQ Devices*, in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Vol. 01 (2023), pp. 506–517.

[99] J. A. Montanez-Barrera, K. Michielsen, and D. E. B. Neira, *Evaluating the performance of quantum processing units at large width and depth*, arXiv:2502.06471.

[100] T. Lubinski, C. Coffrin, C. McGeoch *et al.*, *Optimization applications as quantum performance benchmarks*, ACM Trans. Quantum Comput. **5**, 18 (2024).

[101] Z. He, D. Amaro, R. Shaydulin, and M. Pistoia, *Performance of quantum approximate optimization with quantum error detection*, Commun. Phys. **8**, 217 (2025).

[102] Y. Jin, Z. He, T. Hao *et al.*, *Iceberg beyond the tip: Co-compilation of a quantum error detection code and a quantum algorithm*, arXiv:2504.21172.

[103] D. Forlivesi and D. Amaro, *Flag at origin: A modular fault-tolerant preparation for CSS codes*, arXiv:2508.14200.

[104] A. Y. Kitaev, *Quantum computations: Algorithms and error correction*, Russ. Math. Surv. **52**, 1191 (1997).

[105] C. M. Dawson and M. A. Nielsen, *The Solovay-Kitaev algorithm*, arXiv:quant-ph/0505030.

[106] M. Horodecki, P. Horodecki, and R. Horodecki, *General teleportation channel, singlet fraction, and quasidistillation*, Phys. Rev. A **60**, 1888 (1999).

[107] M. A. Nielsen, *A simple formula for the average gate fidelity of a quantum dynamical operation*, Phys. Lett. A **303**, 249 (2002).

[108] M. Golay, *The merit factor of long low autocorrelation binary sequences (Corresp.)*, IEEE Trans. Inf. Theory **28**, 543 (1982).