Performance of Quantum Approximate Optimization with Quantum Error Detection

Zichang He,[1,*] David Amaro,[2,*] Ruslan Shaydulin,[1] and Marco Pistoia[1]

[1] *Global Technology Applied Research, JPMorganChase, New York, NY 10017, USA*
[2] *Quantinuum, Partnership House, Carlisle Place, London SW1P 1BX, United Kingdom*

(Dated: September 19, 2024)

Quantum algorithms must be scaled up to tackle real-world applications. Doing so requires overcoming the noise present on today's hardware. The quantum approximate optimization algorithm (QAOA) is a promising candidate for scaling up due to its modest resource requirements and documented asymptotic speedup over state-of-the-art classical algorithms for some problems. However, achieving better-than-classical performance with QAOA is believed to require fault tolerance. In this paper, we demonstrate a partially fault-tolerant implementation of QAOA using the $[[k+2, k, 2]]$ "Iceberg" error detection code. We observe that encoding the circuit with the Iceberg code improves the algorithmic performance as compared to the unencoded circuit for problems with up to 20 logical qubits on a trapped-ion quantum computer. Additionally, we propose and calibrate a model for predicting the code performance, and use it to characterize the limits of the Iceberg code and extrapolate its performance to future hardware with improved error rates. In particular, we show how our model can be used to determine necessary conditions for QAOA to outperform Goemans-Williamson algorithm on future hardware. Our results demonstrate the largest universal quantum computing algorithm protected by partially fault-tolerant quantum error detection on practical applications to date, paving the way towards solving real-world applications with quantum computers.

I. INTRODUCTION

Quantum computers are poised to deliver algorithmic speedups for a broad range of application in science and industry [1–3]. However, realizing these speedups requires overcoming the challenge presented by the noise which limits the computational power of today's quantum devices. Error correction [4] provides a scalable path to fault-tolerance and has shown significant progress in hardware recently [5–14]. Nonetheless, quantum error-correction imposes large overheads, making it challenging to execute even small-scale applications fully fault-tolerantly. As a result, fully fault-tolerant demonstrations of quantum algorithms for practical applications have been out of reach of experiment, despite immense progress in implementation and benchmarking of algorithmic components, such as preparation of magic states [15], one-bit addition [16] and quantum Fourier transform [17].

Quantum error detection (QED) codes provide an opportunity for partially fault-tolerant implementation of algorithms in the near term [9, 10, 18–22]. While non-scalable, they can still deliver improved algorithmic performance beyond what is possible without protecting against noise [23–26]. The protection offered by QED codes opens an opportunity to use quantum computers to study the performance of quantum algorithms for sizes and noise rates beyond classical simulation.

The recently proposed $[[k+2, k, 2]]$ "Iceberg" QED code [18] is particularly suitable for near-term algorithms due to its ability to encode expressive circuits, using a universal set of local and global logical rotations, with a low overhead. The Iceberg code has been demonstrated to improve the fidelity of random circuits with up to 8 logical qubits and 1323 physical two-qubit gates [18], the performance of quantum phase estimation with up to 4 logical qubits and 920 two-qubit gates [25], and the fidelity of ground state preparation with probabilistic imaginary-time evolution with 4 logical qubits and up to 906 two-qubit gates [26]. While not fully fault-tolerant, these experiments provided preliminary evidence that for circuits with small numbers of qubits the Iceberg code can improve algorithmic performance.

Quantum Approximate Optimization Algorithm (QAOA) [27, 28] is a quantum optimization heuristic applicable to a broad range of combinatorial optimization problems in finance and other industries [29–32]. QAOA has been shown to provide a quantum algorithmic speedup over state-of-the-art solvers for some problems [33, 34], motivating its implementation on hardware. While relatively low resource requirements enabled QAOA execution on non-error-corrected hardware [35–42], realizing the speedup offered by QAOA is widely believed to require fault tolerance [43, 44]. We remark that both quantum hardware performance and the impact of quantum noise on QAOA have been subject of extensive interest [45–52].

We demonstrate a partially fault-tolerant implementation of QAOA applied to the MaxCut problem on Quantinuum H2-1 trapped-ion quantum computer [54] with the Iceberg code. At the time of experiments, the H2-1 device had 32 all-to-all connected qubits and 99.8% two-qubit gate fidelity [38]. We execute circuits with up to 24 logical qubits encoded into up to 26 physical

* These authors contributed equally to this work. Correspondence should be addressed to zichang.he@jpmchase.com.

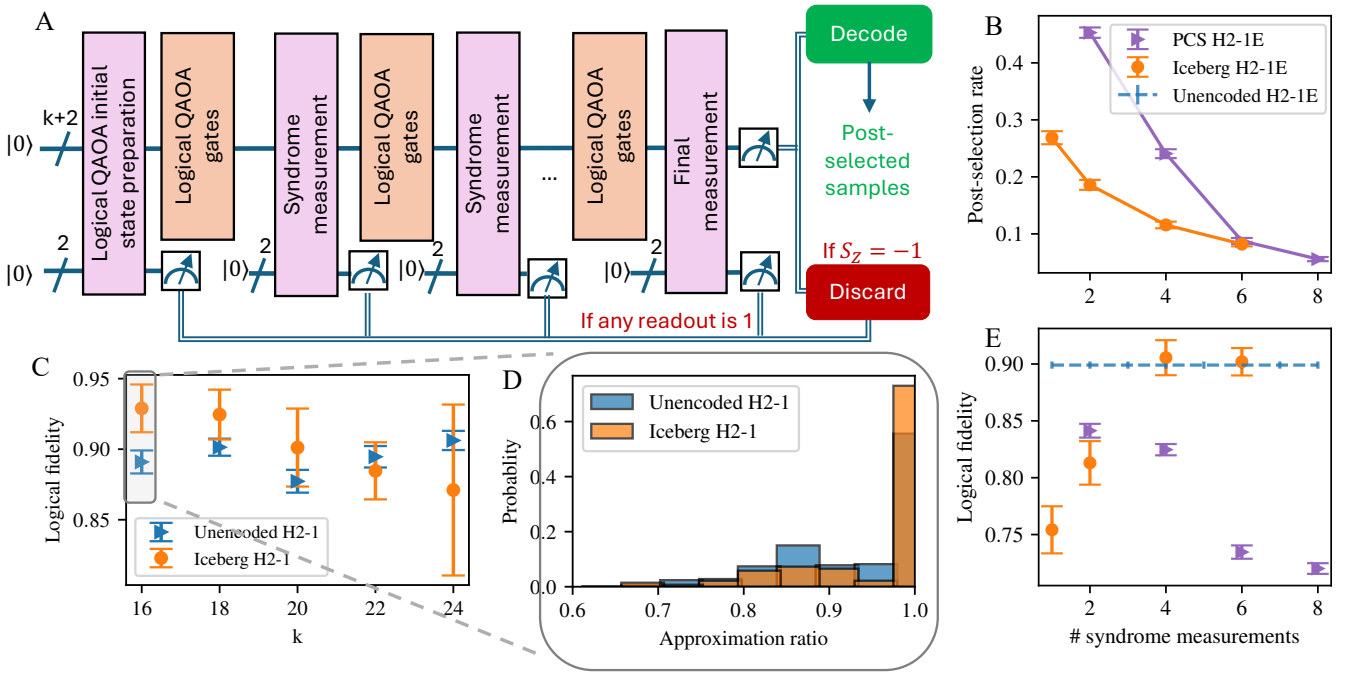

FIG. 1. **Motivation: Iceberg code is a performant method for error detection in the near-term. A** Iceberg code detects errors that occur in execution of an k-qubit circuit by encoding it in $(k+2)$ physical qubits. **B** Shots containing a detected error are discarded, resulting in a post-selection overhead. **C** Performance of QAOA with and without the Iceberg code on the Quantinuum H2-1 quantum computer. Here, the logical fidelity, defined in Eq. (17), directly indicates the approximation ratio. The Iceberg code improves performance on small problems, while being detrimental on larger ones. **D** An example of measured samples with and without the Iceberg code. After detecting the errors, the probability of the higher energy states is amplified, reflecting the approximation to the noiseless QAOA performance. **E** Iceberg code performs better than other commonly-used techniques for error mitigation in QAOA circuits like Pauli Check Sandwiching (PCS) [53]. Error bars show the standard errors.

qubits using up to 813 physical two-qubit gates and observe that protecting the circuit with the Iceberg code leads to improved approximation ratio as compared to the unencoded circuit for problems requiring up to 20 logical qubits. However, we also observe that beyond 20 logical qubits, Iceberg code does not yield an improved algorithmic performance. To the best of our knowledge, these experiments are the largest evaluation of Iceberg code and the largest QED-encoded application demonstration in terms of the number of logical qubits to date.

To understand the protection capability of the Iceberg code, we propose a model that predicts the code performance. The model efficiently constructs an analytical estimation of the logical fidelity and post-selection rate as a function of the circuit size and three error rates related to the noise produced by two-qubit physical gates. We calibrate the model by simulating a large set of QAOA circuits of varying sizes with and without the Iceberg code in the emulator [8, 55, 56] of the H2-1 quantum computer. The calibrated model is then used it to characterize the regimes in which the Iceberg code improves the algorithmic performance of QAOA. Specifically, we identify the ranges for the number of logical qubits, QAOA depth, and the number of syndrome measurements for which Iceberg code is beneficial. Furthermore, our model can be used to predict the performance of the Iceberg code on future improved hardware. We demonstrate conditions on effective error rates for QAOA to outperform the Goemans-Williamson (GW) [57] algorithm on small graphs.

II. BACKGROUND

We begin with a brief review of relevant concepts about QAOA and the Iceberg code.

While our model can be generalized to any optimization problem, in this paper we focus on the MaxCut problem as a commonly-studied benchmark problem.

Given a graph $G(k, E)$ with k vertices and set of edges E, the MaxCut problem consists in finding a cut that partitions the vertices into two sets that maximise the number of edges between them. Cuts can be represented by strings $\boldsymbol{z}$ of k bits with value $z_i = \pm 1$ if vertex i is in one set or the other. The MaxCut objective function can be written as $f(z) = \sum_{(i,j)\in E}(1 - z_i z_j)$.

On qubits, the MaxCut problem is equivalent to finding the ground state of the following k-qubit Hamilto-

nian:

$$\mathcal{H} = \sum_{(i,j)\in E} Z_i Z_j, \tag{1}$$

where we define the Pauli operators as $\mathcal{P} = \{I, X, Y, Z\}$ and Z_i as the Pauli-Z operator acting on qubit i. The k-qubit computational state $|\boldsymbol{z}\rangle = \otimes_{i=1}^{k} |z_i\rangle$, with $Z_i |\boldsymbol{z}\rangle = z_i |\boldsymbol{z}\rangle$, that minimizes the cost Hamiltonian represents the optimal solution of the problem.

A. Quantum Approximate Optimization Algorithm

QAOA is a quantum algorithm for combinatorial optimization. It solves optimization problems by preparing quantum state using a sequence of ℓ layers of alternating cost Hamiltonian and mixing Hamiltonian operators, parameterized by vectors $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ respectively.

$$|\boldsymbol{\psi}\rangle = e^{-i\beta_\ell \mathcal{M}} e^{-i\gamma_\ell \mathcal{H}} \cdots e^{-i\beta_1 \mathcal{M}} e^{-i\gamma_1 \mathcal{H}} |\boldsymbol{\psi}_0\rangle \tag{2}$$

The parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are chosen such that the measurement outcomes of $|\boldsymbol{\psi}\rangle$ correspond to high-quality solutions of the optimization problem with high probability. In this paper, we take the initial state $|\boldsymbol{\psi}_0\rangle = |+\rangle^{\otimes k}$ as the equal superposition of all possible candidate solutions, and the mixing Hamiltonian as a summation of all single-qubit Pauli-X operator $\mathcal{M} = \sum_{i=1}^{n} X_i$.

Denoting the value of optimal cut by $f_{\max}$, we can quantify how well QAOA with state $|\boldsymbol{\psi}\rangle$ solves the MaxCut problem by computing the *approximation ratio*:

$$\alpha(\boldsymbol{\psi}) = \frac{|E| - \langle \boldsymbol{\psi}|\mathcal{H}|\boldsymbol{\psi}\rangle}{2 f_{\max}}. \tag{3}$$

Recent progress in parameter setting heuristics has considerably advanced the execution of QAOA in the early fault-tolerant era [42, 58], with good parameter choices available for many problems. A set of parameters that leads to good approximation ratios was proposed in [59] for the MaxCut problems on regular graphs that we solve in this work. Throughout our paper we use these "fixed angles" to set QAOA parameters in the experiments.

B. Iceberg Code

The Iceberg code protects k (even) logical qubits with $n = k + 2$ physical qubits and two ancillary qubits. We label the physical qubits as $\{t, 1, 2, \ldots, k, b\}$, where the two additional qubits are called *top* t and *bottom* b for convenience. The two code stabilizers and the logical operators are

$$S_X = X_t X_b \prod_{i=1}^{k} X_i, \tag{4}$$

$$S_Z = Z_t Z_b \prod_{i=1}^{k} Z_i, \tag{5}$$

$$\bar{X}_i = X_t X_i \quad \forall\, i \in \{1, 2, \ldots, k\}, \tag{6}$$

$$\bar{Z}_i = Z_b Z_i \quad \forall\, i \in \{1, 2, \ldots, k\}. \tag{7}$$

From these definitions one can see that the logical gates of the QAOA circuit are implemented as the physical gates

$$\exp(-i\beta \bar{X}_i) = \exp(-i\beta X_t X_i), \tag{8}$$

$$\exp(-i\gamma \bar{Z}_i \bar{Z}_j) = \exp(-i\gamma Z_i Z_j). \tag{9}$$

In Quantinuum devices, these physical gates are implemented by just one native two-qubit gate $\exp(-i\theta Z_i Z_j)$ and various single-qubit Clifford gates.

As depicted in Fig. 1A, the Iceberg code employs an initialization block to prepare the initial QAOA state $|\bar{+}\rangle^{\otimes n}$ in the common $+1$ eigenspace of the stabilizers. The logical QAOA gates in Eqs. (8) and (9) are then implemented in blocks, interleaved with syndrome measurement blocks until the QAOA circuit is complete. These syndrome measurement blocks measure the stabilizers regularly across the circuit to prevent the accumulation of noise. The final measurement block measures the stabilizers as well as the k data qubits. The precise form of these blocks is depicted in Appendix A. Accepted samples can be decoded by a classical post-processing and serve as a candidate solutions for the problem.

To detect these errors, the fault-tolerant initialization, syndrome measurement and final measurement blocks, employ two ancillas. In the absence of noise, the state remains purely in the $+1$ eigenspace of the stabilizers during the entire circuit execution and ancillas always output a $+1$ when measured. The final measurement block additionally measures the stabilizer S_Z, which is also expected to be measured as $+1$ in the absence of noise. Therefore, a -1 output in any of them signals the presence of an error caused by noise, and the circuit execution is discarded.

The fault-tolerant design of the initialization, syndrome measurement and final measurement blocks ensures that no single faulty component in these blocks (like a two-qubit gate) can cause a logical error. In contrast, our logical gates, despite being natural for the hardware, are not fault-tolerant, as some errors in their physical implementation can not be detected. We nevertheless show in Sec. III B 2 that undetectable errors are rare, rendering the QAOA protection of the Iceberg code effectively fully fault-tolerant.

III. RESULTS

We now present our results. First, we summarize the results obtained on the hardware. Then, we discuss the model fitting results and the performance predictions on future hardware. The H-series emulators [8, 55, 56] we use perform a state-vector simulation where noise is randomly sampled following realistic noise models and then inserted into the circuit. Currently, the most influential noise channels are gate errors and single-qubit coherent dephasing from memory errors. The performance gap between the hardware and emulator experiments is discussed in Appendix B for completeness.

A. Iceberg code protection of QAOA on hardware

The performance of the Iceberg code with QAOA on 3-regular graph MaxCut on the Quantinuum H2-1 quantum computer [54] is shown in Fig. 1C. The logical fidelity reported in this figure is estimated from the average energy measured experimentally by assuming a global white noise model distribution, as described in Sec. V C. We fix the QAOA depth to $\ell = 10$ and vary the number of logical qubits k, randomly selecting one MaxCut graph instance per k. For each problem we run QAOA unencoded and QAOA protected by the Iceberg code with three intermediate syndrome measurements. Throughout this paper, the final measurement is counted as a syndrome measurement, so in the previous experiment we say that four syndrome measurements are used. The Iceberg data has larger error bars due to the smaller number of post-selected samples.

The histogram in Fig. 1D reports the hardware shots of several Iceberg and unencoded QAOA circuits for $k = 16$. After post-selection, the output distribution has higher weight on bitstrings with higher approximation ratio, as expected from a better protection against noise.

We compare the performance with that of the Pauli check sandwiching (PCS) [53, 60], an error detection scheme with a similar motivation to that of the Iceberg code. PCS uses pairs of parity checks to detect some but not necessarily all errors that occur in a given part of the circuit. The parity checks are chosen based on the symmetries already present in the circuit. For QAOA circuits considered in this work, the problem Hamiltonian commutes with $X^{\otimes k}$ and $Z^{\otimes k}$, so we use them as the checks of our PCS experiments. To unify the notation with Iceberg code, a pair of $X^{\otimes k}$ and $Z^{\otimes k}$ checks is denoted as one syndrome measurement. For example, one syndrome measurement in PCS means that we select one cost Hamiltonian layer $e^{-i\gamma\mathcal{H}}$ and sandwich it with two parity checks. The overhead of one syndrome measurement in PCS includes an additional $4k$ two-qubit gates, along with two ancillas. The comparison with PCS on a $k = 18$, $\ell = 11$ QAOA circuit is shown in Fig. 1B and E, where all data are from the H2-1 emulator (H2-1E). We observe that PCS leads to a lower logical fidelity that does not increase with the number of syndrome measurements. At this scale, the large overhead and the non-fault tolerant design of the PCS method decreases the circuit performance. At the same time, we observe that the Iceberg code can effectively improve the QAOA performance and obtain a higher logical fidelity than the unencoded circuit with four syndrome measurements.

B. Estimated performance from our model

To understand the protection capability of Iceberg code, we propose the performance model of Sec. V B for the unencoded and Iceberg code circuits. The model outputs analytical functions of the logical fidelity $\mathcal{F}_{\text{une}}$ for the unencoded circuits, the logical fidelity $\mathcal{F}_{\text{ice}}$ for the Iceberg code circuits, and the post-selection rate $1 - D$ for the Iceberg code (D is the discard rate). Inputs from the circuit are the numbers of logical qubits k, logical single-qubit gates g_1, logical two-qubit gates g_2, and syndrome measurements s. From the hardware, the model for the unencoded and the Iceberg code respectively inputs only one and three error rates related to the noise produced by two-qubit physical gates. These are motivated and described in more detail in the next section. In this work we leave the error rates as fitting parameters so that, when fitted to data from the H2-1 emulator the fitted values incorporate corrections from other noise sources.

Since QAOA experiments on hardware or emulators output the approximation ratio instead of the logical fidelity, we extend the performance model by approximating the noise distribution as that of a global white noise. Details are discussed in Sec. V C.

1. *Dataset and model validation*

We use the emulator of the Quantinuum H2-1 quantum computer to generate a dataset with varying number of logical qubits in the range $k \in [8, 26]$, QAOA layers in the range $\ell \in [1, 11]$, and syndrome measurements in the range $s \in [1, 8]$. This dataset contains 115 circuits for the Iceberg code and 56 for the unencoded circuits. We take 1000 shots for the unencoded circuits and 3000 shots for the Iceberg code circuits before post-selection.

From this dataset we select partial data that have relatively stable logical fidelity and large numbers of two-qubit gates to fit the model. For the Iceberg code model,

	Iceberg			Unencoded
Parameter	p_{cx}	p_c	p_a	p_ℓ
Emulator	1.28e-3	[1.3e-05, 3.2e-05]	[4.3e-4, 1.1e-3]	[4.7e-4, 1.0e-3]
Model	5.5e-3	7.0e-5	2.2e-3	4.4e-4
CI	[5.0e-3, 6.2e-3]	[4.3e-5, 9.5e-4]	[1.9e-3, 2.5e-3]	[4.0e-4, 5.0e-4]

TABLE I. Error rates from the H2-1 emulator [8, 55, 56] and the performance model.

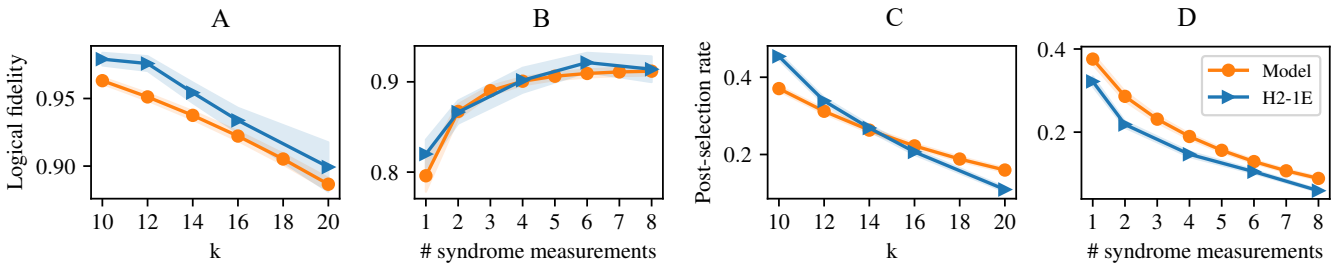

FIG. 2. **Proposed model accurately reflects behavior of Iceberg encoding circuits observed in high-accuracy emulation.** The fitted model matches the qualitative and quantitative behavior of logical fidelity and post-selection rate for both varying qubit count with a fixed $\ell = 9$ (**A,C**) and varying number of syndrome measurements with a fixed $k = 16$ and $\ell = 11$ (**B,D**). The shaded regions represent the standard errors.

that has three fitting parameters, we use data from 64 encoded circuits while the data from 15 unencoded circuits is used to fit the unencoded model, that has only one fitting parameter. We additionally filter out those Hamiltonian terms of every QAOA circuit whose expected values are outliers with respect to the white noise approximation. More details are provided in Appendix B. The mean and 95% confidence interval of the fitted parameters from 3000 boostrapping iterations are reported in Table I.

To validate the accuracy of the performance model, we present the logical fidelity and post-selection rate of experimental data alongside the model predictions in Fig. 2. We find that our model can match the experimental results both qualitatively and quantitatively. The model and experiment fidelities and post-selection rates for every selected circuit, as well as the deviations from the white noise simplification are presented in Appendix B.

2. *Fitted error rates*

The deviations between the fitted error rates and the emulator noise rates presented in Table I contain valuable information about where other noise sources accumulate.

Starting from the error rate p_{cx} of CNOT gates, the fitted value is more than four times larger than the emulator error rate, showing that a significant amount of noise unaccounted by the performance model accumulates in the error detection blocks of the Iceberg code. For the logical gates we consider two noise channels, with error rates p_c and p_a, that introduce Pauli errors that commute and anti-commute, respectively, with the two Iceberg code stabilizers. Since the values established in the emulator depend on the rotation angles of the logical gates and the QAOA circuits do not present a clear tendency towards any particular angle, Table I presents the minimum and maximum values. We find that these fitted error rates are almost twice larger than the maximum value given by the emulator, hinting again that the logical gates in the Iceberg code circuit accumulate unnacounted noise. In contrast, the fitted value of the error rate p_ℓ of the unencoded logical gates is well approximated by the minimum value established by the emulator.

Importantly, the only single errors that can cause a logical error in the Iceberg code logical gates happen with the smallest probability $p_c \sim$7e-5 among the three noise sources. This indicates that for circuits with a small number of logical gates, the Iceberg code effectively behaves as a fully fault-tolerant quantum error detection code.

The values reported for the emulator in Table I are obtained from the parameters of the depolarising channel for the native two-qubit gates $\exp(-i\theta ZZ)$ and their dependence with the QAOA rotation angles $\theta \in \{\gamma, \beta\}$. The depolarising channel assigns a probability q_σ to each of the 15 Pauli errors in $\sigma = \mathcal{P}^{\otimes 2} \setminus \{I^{\otimes 2}\}$ after the native gate. The total probability of a Pauli error is the value of p_{cx} reported in the table. These values are corrected by a multiplicative factor defined as linearly increasing function $r(\theta) \simeq a + b|\theta|$ in the angle magnitude, such that $r(\pi/4) = 1$ for maximally entangling gates like the CNOT. The error rate of unencoded logical gates is then $p_\ell(\theta) = p_{cx} r(\theta)$. For the Iceberg code we additionally separate the error rate of commuting errors $q_c = q_{X^{\otimes 2}} + q_{Y^{\otimes 2}} + q_{Z^{\otimes 2}}$ from that of anti-commuting errors $q_a = p_{cx} - q_c$. To unify with the model error rates, we factorise this single channel into a product of a commuting channel with error rate $p_c(\theta) = q_c r(\theta)(1 - q_a r(\theta))$ and an anti-commuting channel with error rate $p_a(\theta) = q_a r(\theta)$. We report the minimum and maximum among all QAOA rotation angles.

3. *Frontiers of the Iceberg Code performance.*

Next, we analyze the performance of the Iceberg QAOA circuits based on the fitted model. We report the difference $\mathcal{F}_{\text{ice}} - \mathcal{F}_{\text{une}}$ in logical fidelity between Iceberg and unencoded circuits in Fig. 3A, and the post-selection rate of Iceberg circuits $1 - D$ in Fig. 3C, for varying numbers of syndrome measurements.

As observed from the shift of the breakeven frontiers (red lines) in Fig. 3A, the QAOA logical fidelity stabilizes as the number of syndrome measurements increases, even though the post-selection rate, indicated by the orange

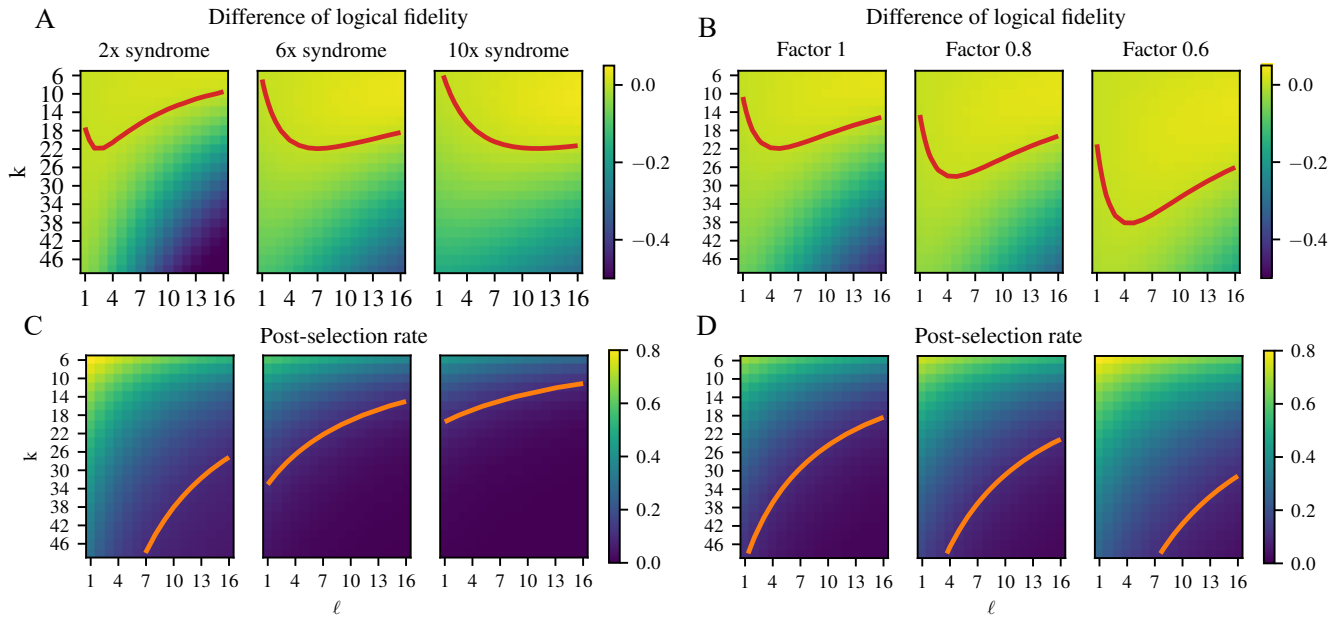

FIG. 3. **Model prediction**: Predicting the performance of QAOA with the number of logical qubits in the range $k \in [6, 48]$ and the number of QAOA layers in the range $\ell \in [1, 16]$. We use the model proposed in this work to estimate (**A, B**) the difference $\mathcal{F}_{\text{ice}} - \mathcal{F}_{\text{une}}$ in logical fidelity between the Iceberg and the unencoded circuits, and to estimate (**C, D**) the post-selection rate. In **A** and **C**, we use the model fitted error rates in Table I and vary the number of syndrome measurements. In **B** and **D**, we fix the number of syndrome measurements at 4 and scale down the model error rates by the indicated factors. The red lines in the top row (**A**, **B**) show where the logical fidelity of Iceberg code circuits equals that of unencoded circuits. The orange lines in the bottom row (**C**, **D**) indicate where the Iceberg code circuits have a 10% post-selection rate.

lines in Fig. 3C, decreases. This aligns with our experimental data and findings in the literature [18], which suggest that the initial syndrome measurements significantly enhance circuit performance, while the marginal gains diminish with an increasing number of syndrome measurements.

4. *Predicting performance on future hardware*

We now use our model to predict the performance of QAOA with the Iceberg code on future quantum hardware. To study this, we extrapolate the model performance by scaling all the model parameters in Table I by a varying factor. A smaller factor corresponds to smaller effective error rates, indicating higher fidelity of the quantum hardware. Scaling all error rates down by the same factor is clearly an additional simplification, as hardware development will not necessarily reduce all noise sources homogeneously and at the same pace. Nevertheless, this analysis provides a valuable qualitative perspective on the potential performance in a foreseeable scenario.

As shown in Fig. 3B and D, as the factor decreases, we observe a significant shift of the performance frontier to larger number k of logical qubits while the post-selection rate improves dramatically. This indicates that with higher-quality quantum hardware, we can push the breakeven frontier of logical fidelity to deeper circuits on larger problem instances with less post-selection overhead.

To elucidate the conditions for QAOA to become competitive with classical solvers, we use our model to answer the question of when a QAOA hardware experiment can outperform the Goemans-Williamson (GW) algorithm [57] in terms of approximation ratio. As reported in the literature [59], a noiseless QAOA with fixed parameters has been able to surpass the GW algorithm on small graphs. In Fig. 4A, we show an example of solving $k = 16$-node 3-regular graphs with noiseless QAOA, GW, and Iceberg QAOA with four syndrome measurements as well as unencoded QAOA emulated on the H2-1 emulator. The noiseless QAOA is able to outperform GW for $\ell \geq 6$ layers. However, both the Iceberg and unencoded QAOA have not yet surpassed GW.

Specifically, at $\ell = 10$, the approximation ratio of noiseless QAOA is $0.9810\ldots$, while the average approximation ratio for GW is $0.9554\ldots$. This implies that the logical fidelity of a noisy QAOA must be of at least $0.9554/0.9810 \simeq 0.974$ to outperform GW, assuming our white noise model approximation. In Fig. 4B, we vary the scaling factor of the model parameters to determine when this breakeven logical fidelity can be achieved. The results indicate that the Iceberg and unencoded circuits require scaling factors of approximately 0.81 and 0.60, re-

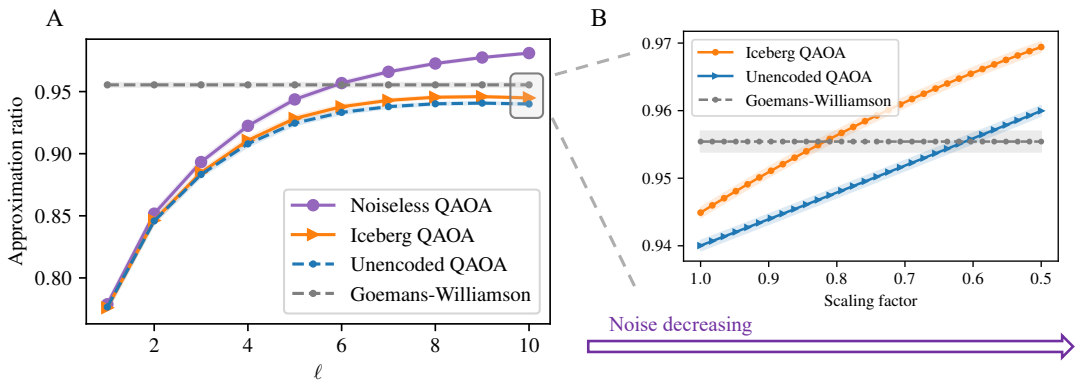

FIG. 4. **Hardware improvement necessary for QAOA to become competitive with the Goemans-Williamson algorithm. A** Solve $k = 16$ MaxCut using different solvers. Each data is reported as the mean of approximation ratio over 100 $k = 16$ 3-regular graphs. The standard errors are too small to be seen. **B** Scaling of model parameter to beat Goemans-Williamson (GW) algorithm for $k = 16$ graphs. The Iceberg code helps the QAOA to beat GW earlier than an unencoded one.

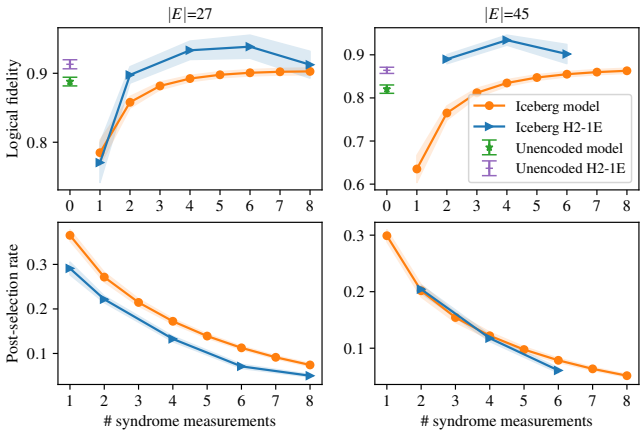

FIG. 5. Comparison between experimental data and model predictions on random Erdős–Rényi graphs with different numbers of edges. The prediction of Iceberg logical fidelity is less accurate compared to testing on 3-regular graphs, while the prediction of unencoded logical fidelity and the prediction of post-selection rate remain accurate.

spectively. This suggests that the Iceberg code enables a breakeven fidelity on small MaxCut problems much earlier than unencoded circuits as hardware technology advances.

5. *Model generalization beyond 3-regular graphs*

So far, we have used 3-regular graphs to fit the performance model and analyzed the model's performance on extrapolated 3-regular graphs. To test the generalizability of the model, we validate it on random Erdős–Rényi graphs, which have different topologies compared to 3-regular graphs. We fix the number of nodes at $k = 18$, set the number of edges as $|E| \in \{27, 45\}$ with all edges generated randomly, and select one graph for each of the two sizes.

We present the comparison between experimental results and model predictions in Fig. 5. The model predictions for the logical fidelity of the Iceberg code circuits are less accurate, specially with the densest graph of $|E| = 45$ edges. The model predictions for the unencoded logical fidelity and the Iceberg code post-selection rates are comparably more accurate. This indicates that the fitted model works well for problems with similar topologies, but highlights the limitation of the model's generalization to different problems. We suspect that the worse model performance on these graphs is caused by the different amount of unaccounted noise accumulated in the comparably deeper circuits for these graphs.

IV. DISCUSSION

We demonstrate that the post-selected samples with the Iceberg code present a higher approximation ratio than the samples from unencoded circuits. This allows us to study the performance of QAOA in an effective noise regime closer to the noiseless computation; at least for circuits on the beneficial side of the breakeven frontier. We can see the current breakeven frontier of the Iceberg QAOA on 3-regular graphs in Fig. 3A. For example, for 6 syndrome measurements, the breakeven frontier is up to $k = 20$ logical qubits for $\ell \in [4, 12]$ of QAOA layers. However, there are multiple opportunities to improve this result and achieve improved performance with Iceberg code as compared to the unencoded circuits for larger qubit counts.

First, other problems may be more ameanable to Iceberg code. Specifically, dealing with sparse-graph problems, like the MaxCut of 3-regular graphs discussed in this paper, is not particularly beneficial for the Iceberg code compared to fully-connected Hamiltonians like the

Sherrington-Kirkpatrick model. This is because the Iceberg code can execute two-qubit logical rotations with no overhead, whereas one-qubit logical rotations require a noisier two-qubit physical gate. Iceberg QAOA could be more advantageous for dense graph problems where the number of two-qubit $Z^{\otimes 2}$ terms in the problem Hamiltonian is much larger than the number of single-qubit X terms in the mixing Hamiltonian.

Second, the compilation of Iceberg circuits could be further optimized. We speculate that the high deviations of the error rates observed between the emulator and fitted error rates in Table I could be explained by a larger amount of memory noise accumulated in the highly sequential syndrome measurement blocks and the QAOA mixing layer than in the optimized unencoded circuits. Currently, we are using pytket [61] to compile the logical gates alone, but a better strategy that jointly compiles the logical gates and the error detection blocks of the Iceberg code could potentially improve hardware performance while reducing execution time.

While performance extrapolation indicates promising results with improved hardware fidelity, we observe that, with fixed model parameters, increasing the number of syndrome measurements marginally diminishes the performance gains in the extrapolation heatmaps. Additionally, the overhead of post-selecting samples grows rapidly. This observation is consistent with the experimental results on the H2-1 emulator, indicating that the protective power of the Iceberg error detection code is limited. This reinforces the need for quantum error correction to achieve error rates low enough to run large-scale circuits.

V. METHODS

This section provides further details on the experimental realization, presents the model construction, and details the fitting to experimental data.

A. Location of syndrome measurements

As depicted in Fig.1A , in this work we place syndrome measurements evenly spaced in the circuit so that every block of logical gates has roughly the same number of logical gates.

In Fig. 6, we provide evidence supporting this strategy. The $k = 16, \ell = 11$ logical QAOA circuit is evenly partitioned into eight blocks of logical gates and we introduce a single syndrome measurement in the seven intermediate positions between them. By comparing the logical fidelity obtained from these seven circuits, we find that the circuit with the syndrome measurement inserted in the middle (labeled 4) presents the highest fidelity, at the cost of the smallest post-selection rate.

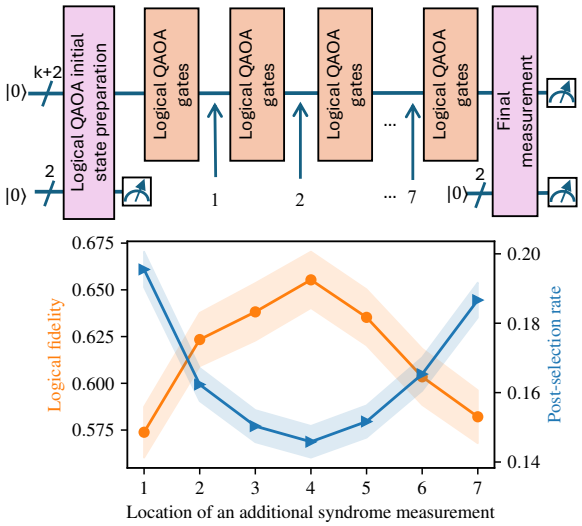

FIG. 6. Top: Circuits with different locations for a single syndrome measurement, labeled 1-7. Bottom: Performance of the different circuits. The circuit with the syndrome measurement in the middle (labeled 4) detects the most errors and achieves the best logical fidelity. All data include the final measurement and fault-tolerant initialization. Without any syndrome measurement, the logical fidelity is 0.5128 ± 0.0115 and the post-selection rate is 0.2762 ± 0.0058. With one additional syndrome measurement, all circuits outperform the one without the syndrome measurement.

B. Performance model

This section introduces the detailed model to predict the performance of the unencoded and Iceberg code circuits. To build the model outputs efficiently, we consider the following noise channels:

- Uniform two-qubit depolarizing channel with error rate p_ℓ after every two-qubit logical gate of the unencoded circuit: insert a random Pauli error from the set $\mathcal{P}^{\otimes 2} \setminus \{I^{\otimes 2}\}$ on the gate support.

- Uniform two-qubit depolarizing channel with error rate p_{cx} after every two-qubit CNOT gate in the error detection blocks of the Iceberg code circuit: insert a random Pauli error from the set $\mathcal{P}^{\otimes 2} \setminus \{I^{\otimes 2}\}$ on the gate support.

- A noise channel with error rate p_c that introduces a random error that commutes with both stabilizers after every logical gate of the Iceberg code: insert a random Pauli error from the set $\{X^{\otimes 2}, Y^{\otimes 2}, Z^{\otimes 2}\}$ on the gate support.

- A noise channel with error rate p_a that introduces a random error that anti-commutes with both stabilizers after every logical gate of the Iceberg code: insert a random Pauli error from the set $\mathcal{P}^{\otimes 2} \setminus \{I^{\otimes 2}, X^{\otimes 2}, Y^{\otimes 2}, Z^{\otimes 2}\}$ on the gate support.

Overall, we have one error parameter p_ℓ for unencoded circuits, and three error parameters, p_{cx}, p_c, and p_a for Iceberg circuits. The probabilities of no error P_0, one error P_1, and two or more errors P_2 in a set of g gates with error rate p are respectively defined as

$$P_0(p,g) = (1-p)^g, \tag{10}$$
$$P_1(p,g) = gp(1-p)^{g-1}, \tag{11}$$
$$P_2(p,g) = 1 - P_0(p,g) - P_1(p,g). \tag{12}$$

We additionally make two simplifications to construct the model outputs efficiently. The first one states that

Simplification 1 *every undetected error produces a logical error, unless the fault-tolerance of the Iceberg code prevents it.*

For unencoded circuits, since no error is detectable, every error produces a logical error by virtue of this simplification, so we model the *unencoded logical fidelity* as the probability of absolutely no error:

$$\mathcal{F}_{\text{une}} = P_0(p_\ell, g_2). \tag{13}$$

From the fault-tolerant design every single error in the error detection blocks of the Iceberg code is either detectable or acts trivially, so none of these single errors contribute to the logical infidelity of the Iceberg code. However, two or more errors can produce undetectable errors, that, by virtue of this simplification, produce a logical error. The usually implicit justification of this simplification is that the number of errors that act trivially in the circuit is exponentially smaller in the circuit size than the number of undetectable errors that act non-trivially at the logical level.

1. Model for the Iceberg code

The model for the Iceberg code incorporates the effects of error detection and fault-tolerance. To construct the model efficiently we divide the circuit into blocks of initialization, logical gates, syndrome measurement and final measurement, respectively, as depicted in Fig. 1. At every block we consider the probabilities of errors causing

1. a *harmless error* H if they excite no ancilla and no stabilizer and act trivially on the state, like a Z error before the measurement of an ancilla,
2. a *logical error* L that excites no ancilla and no stabilizer but acts non-trivially on the state, like a two-qubit $X^{\otimes 2}$ error after a logical gate,
3. an *exciting error* E that excites a stabilizer but not an ancilla, propagating such error to the next block without an immediate discard, like a single-qubit X error on a code qubit inserted by the last CNOT that acts on that qubit in a syndrome measurement,
4. or a *discarding error* D that excites an ancilla, causing an immediate discard, like a single-qubit X error before the measurement of an ancilla.

In the absence of noise the state remains in the $+1$ eigenspace of the stabilisers and ancillas. An error that excites some of them brings the state to their -1 eigenspace, and making the error detectable. We say that exciting and discarding errors are both detectable errors, while harmless and logical errors are undetectable. The sum of the four probabilities adds up to 1 at every block.

These probabilities are initialized from the initialization block, and iteratively updated for every block of logical gates, syndrome measurement, and the final measurement. For example, when adding a block of g logical gates that can suffer internal errors, the input probability of harmless errors H from previous blocks updates to the joint probability $H \leftarrow HP_0(p_c,g)P_0(p_a,g)$ of an input harmless error and no internal errors. At the end of this iterative process we obtain analytical functions for the probabilities of the entire circuit. From those we can compute the model outputs for the Iceberg code.

To construct the model outputs efficiently we make the following additional simplification:

Simplification 2 *Errors are evenly distributed across all possible excitation events. That is, for every number $\mu = 1$ and $\mu \geq 2$ of errors in a block with $m \geq 0$ ancillas the probability of all excitation events is the same and equal to $1/2^{m+2}$. The excitation events are the 2^{m+2} possible ways to excite (or not) the m ancilla(s) and the two stabilizers.*

For example, in the syndrome measurement block there are 16 possible excitation events depending on which of the two stabilizers and the two ancillas are excited or not by errors. Therefore, we assume that $1/16$ of the errors excite no ancilla and no stabilizer, $12/16$ of them excite an ancilla, and $3/16$ excite a stabilizer without exciting the ancillas. This simplification allows to incorporate the error detection and fault-tolerance properties of the Iceberg code in a very natural way. Appendix C show the deviations between the model predictions and the exact fractions computed for small Iceberg code instances.

Initialization block. The top part of Table II presents the contribution of every possible error in this block. For

# errors	0	1	≥ 1	≥ 1	≥ 2
excited ancilla	no	no	no	yes	no
excite stabilizers	no	no	yes	any	no
fraction of errors	1	$1/8$	$4/8$	$3/8$	$1/8$
contribute to	H	H	D	E	L
Initialize the circuit probabilities as	$H \leftarrow P_0(p_{cx}, n+3) + 1/8 P_1(p_{cx}, n+3)$ $L \leftarrow 1/8 P_2(p_{cx}, n+3)$ $E \leftarrow 3/8 P_1(p_{cx}, n+3) + 3/8 P_2(p_{cx}, n+3)$ $D \leftarrow 1/2 P_1(p_{cx}, n+3) + 1/2 P_2(p_{cx}, n+3)$				

TABLE II. Classification of errors in the initialization block.

input error	H	H	H	H	H	L	L	L	L	E	E	E	E	E
# anti-commuting errors	0	0	1	≥ 2	≥ 2	0	1	≥ 2	≥ 2	0	1	1	≥ 2	≥ 2
# commuting errors	0	≥ 1	≥ 0	≥ 0	≥ 0	≥ 0	≥ 0	≥ 0	≥ 0	≥ 0	≥ 0	≥ 0	≥ 0	≥ 0
excited stabilizers	no	no	yes	no	yes	no	yes	no	yes	yes	no	yes	no	yes
fraction of errors	1	1	1	$1/4$	$3/4$	1	1	$1/4$	$3/4$	1	$3/9$	$6/9$	$3/12$	$9/12$
contribute to	H	L	E	L	E	L	E	L	E	E	L	E	L	E
Update rules to add this block	$H \leftarrow HP_0(p_c,g)P_0(p_a,g)$ $L \leftarrow H(1-P_0(p_c,g))P_0(p_a,g) + LP_0(p_a,g) + 1/3EP_1(p_a,g) + 1/4(H+L+E)P_2(p_a,g)$ $E \leftarrow EP_0(p_a,g) + (H+L+2/3E)P_1(p_a,g) + 3/4(H+L+E)P_2(p_a,g)$ $D \leftarrow D$													

TABLE III. Classification of errors in the block of logical gates

each number $\mu = 1$ and $\mu \geq 2$ of errors indicated in the first row (columns 2, 3, 4 and columns 3, 4, 5, respectively), the table distinguishes the possible excitations these errors can cause on the ancilla (second row) and stabilizers (third row). Using the simplification 2 we compute the fraction of errors that contribute to each of the possible events and present them in the fourth row. The fifth row shows how this classification incorporates the fault-tolerance of the initialization block. Every single error is either harmless or detectable, while two or more errors that pass undetected (last column) produce a logical error by virtue of simplification 1. Aggregating the contributions provides the initialization of the circuit probabilities presented in the last rows.

Block of logical gates. To add a block of logical gates we need to update the circuit probabilities from the combination of the input errors from the previous blocks with the internal errors of this block. Since no discard is possible at the block of logical gates, the discard probability D is preserved. The top part of Table III presents the classification of input, internal anti-commuting and internal commuting errors. The fraction of errors is calculated again using the simplification 2. For undetectable input errors, internal two or more anti-commuting errors cause 4 possible excitation events depending on which of the two stabilizers are excited or not. In contrast, input exciting errors and single anti-commuting errors, each create 3 possible stabilizer excitations: only S_Z excited, only S_X excited, or both excited. Together they create 9 possible excitation events. In 3 of them the excitations cancel out (11th column) while 6 preserve some excitation (12th column). Similarly, two or more anti-commuting errors create 4 possible excitation events, depending on which of the two stabilizers get excited or not, so here 12 events are possible when combined with an input exciting error. The bottom part of the table presents the resulting update rules of the circuit probabilities to add a block of g logical gates.

Syndrome measurement block. Similarly to the block of logical gates, the contributions from every combination of input and internal errors are summarized in Appendix C. The main difference is that the circuit is immediately discarded if at least one of the two ancillas is excited.

Final measurement block. To add the final measurement block note that the stabilizer S_Z is computed in post-process while S_X is measured by the ancilla. Note also that the second ancilla of the block is a flag qubit, that can be excited only by internal errors, but not by exciting input errors. Contributions are summarized in Appendix C.

Finally, after adding all blocks one by one, we obtain the model output analytical expressions for the circuit probabilities H, L, D (and the trivial $E = 0$). The *post-selection rate* is $1 - D$ and the *Iceberg code logical fidelity* is

$$\mathcal{F}_{\text{ice}} = H/(1-D). \tag{14}$$

We check in Appendix C that the first and second order terms at low error rates satisfy the expected behaviour

$$1 - \mathcal{F}_{\text{ice}} = (g_1+g_2)p_c + O\left(p_{cx}^2 + p_a^2 + p_{cx}p_a + p_c^2\right). \tag{15}$$

This confirms numerically the partially fault-tolerant nature of the Iceberg code: the only single errors capable of causing a logical error are commuting errors in the logical gates, but no single error from the other two noise sources can cause a logical error.

C. Approximation ratio and logical fidelity

To relate the logical fidelity and the noisy approximation ratio we consider that the circuit noise takes the form of a global white noise channel [62]

$$\boldsymbol{\rho} = \mathcal{F}\left|\psi\right\rangle\left\langle\psi\right| + (1-\mathcal{F})2^{-k}I^{\otimes k}. \tag{16}$$

We can then estimate the noisy approximation ratio from the modeled logical fidelity $\mathcal{F} \in \{\mathcal{F}_{\text{une}}, \mathcal{F}_{\text{ice}}\}$ as

$$\alpha(\boldsymbol{\rho}) = \frac{|E| - \mathcal{F}\left\langle\psi\right|\mathcal{H}\left|\psi\right\rangle}{2f_{\max}}. \tag{17}$$

This is the approximation ratio estimated from the performance model that we report in this work.

Moreover, we sometimes use the white noise channel to estimate the logical fidelity from the samples obtained by running the unencoded circuits and the Iceberg code circuits on hardware or the emulator. In this scenario we have access to the experimental approximation ratio,

or equivalently, to the average energies $\langle \mathcal{H} \rangle_c$ for every QAOA circuit $c \in C_{\text{une}} \cup C_{\text{ice}}$ considered in this work, unencoded or protected by the Iceberg code. The estimated logical fidelity is the one that reproduces such average energy under the white noise channel:

$$\mathcal{F}_c = \frac{\langle \mathcal{H} \rangle_c}{\langle \boldsymbol{\psi} | \mathcal{H} | \boldsymbol{\psi} \rangle}. \tag{18}$$

When we have access to the experimental implementation of the QAOA circuits on hardware or on the emulator this is the estimated logical fidelity we report in this work.

Given a graph with edges E_c solved by the QAOA circuit c, we additionally consider the set of ratios

$$F_c = \left\{ \frac{\langle Z_i Z_j \rangle_c}{\langle \boldsymbol{\psi} | Z_i Z_j | \boldsymbol{\psi} \rangle} \, : \, (i,j) \in E_c \right\} \tag{19}$$

obtained from the experimental expected values $\langle Z_i Z_j \rangle_c$. Note that in general these experimental ratios are not expected to take the same value for all edges, but under the white noise channel, they all equal the channel fidelity.

To quantify the deviation between the experimental data and the white noise simplification, we consider the normalized distance between the experimental logical fidelity and the ratios as

$$d\left(F_c, \mathcal{F}_c\right) = \frac{1}{|E_c|} \sqrt{l\left(F_c, \mathcal{F}_c\right)}, \text{with} \tag{20}$$

$$l\left(F_c, \mathcal{F}_c\right) = \sum_{\mathcal{F}_{ij} \in F_c} \left(\mathcal{F}_{ij} - \mathcal{F}_c\right)^2. \tag{21}$$

D. Model fitting

For the unencoded circuits, we use the least squares method as a loss function to minimize the residual between the model logical fidelity $\mathcal{F}_{\text{une}}$ and the experimental ratios obtained from the unencoded circuits. For every QAOA instance c, we compute the set of ratios $F_{c,\text{une}}$ and then aggregate them all to the loss function:

$$l_{\text{une}}(\mathcal{F}_{\text{une}}) = \sum_{c \in C_{\text{une}}} l(F_c, \mathcal{F}_{\text{une}}). \tag{22}$$

For fitting the Iceberg code model to the experimental data, still leveraging the least squares method, we minimize the residuals of both the logical fidelity and discard rate. Given a QAOA instance $c \in C_{\text{ice}}$ with experimental discard rate D_c we define the loss function for multiple QAOA instances as

$$l_{\text{ice}}(\mathcal{F}_{\text{ice}}) = \sum_{c \in C_{\text{ice}}} \frac{1}{|F_c|} l(F_c, \mathcal{F}_{\text{ice}}) + (D_c - D)^2. \tag{23}$$

ACKNOWLEDGMENTS

We would like to thank the experimental scientists in the Quantinuum team that design and maintain the H-series devices. We extend our gratitude to Drs. Matthew DeCross, Ciaran Ryan-Anderson and Selwyn Simsek for valuable discussions about the modeling of memory errors in these devices and Drs. David Hayes, Changhao Li, Pradeep Niroula, and Shree Hari Sureshbabu for the preliminary statistical analysis of the numerical experiments. We appreciate the comments and suggestions of Drs. Chris N. Self, Michael Perlin and Sivaprasad Omanakuttan on the manuscript. We thank the technical staff at JPMorganChase's Global Technology Applied Research Center for their support and helpful discussions.

[1] A. M. Dalzell, S. McArdle, M. Berta, P. Bienias, C.-F. Chen, A. Gilyén, C. T. Hann, M. J. Kastoryano, E. T. Khabiboulline, A. Kubica, *et al.*, Quantum algorithms: A survey of applications and end-to-end complexities, arXiv preprint arXiv:2310.03011 (2023).

[2] D. Herman, C. Googin, X. Liu, Y. Sun, A. Galda, I. Safro, M. Pistoia, and Y. Alexeev, Quantum computing for finance, Nature Reviews Physics **5**, 450 (2023).

[3] Z. He, S. Chakrabarti, D. Herman, N. Kumar, C. Li, P. Minssen, P. Niroula, R. Shaydulin, Y. Sun, S. H. Sureshbabu, *et al.*, Invited: Challenges and opportunities of quantum optimization in finance, in *2024 61th ACM/IEEE Design Automation Conference (DAC)* (IEEE, 2024) pp. 1–4.

[4] D. Gottesman, Surviving as a quantum computer in a classical world, Textbook manuscript preprint (2016).

[5] R. Acharya, I. Aleiner, R. Allen, T. I. Andersen, M. Ansmann, F. Arute, K. Arya, A. Asfaw, J. Atalaya, R. Babbush, *et al.*, Suppressing quantum errors by scaling a surface code logical qubit, Nature **614**, 676–681 (2023).

[6] R. Acharya, L. Aghababaie-Beni, I. Aleiner, T. I. Andersen, M. Ansmann, F. Arute, K. Arya, A. Asfaw, N. Astrakhantsev, J. Atalaya, *et al.*, Quantum error correction below the surface code threshold, arXiv preprint arXiv:2408.13687 (2024).

[7] Q. Xu, J. P. Bonilla Ataides, C. A. Pattison, N. Raveendran, D. Bluvstein, J. Wurtz, B. Vasić, M. D. Lukin, L. Jiang, and H. Zhou, Constant-overhead fault-tolerant quantum computation with reconfigurable atom arrays, Nature Physics **20**, 1084–1090 (2024).

[8] C. Ryan-Anderson, J. G. Bohnet, K. Lee, D. Gresh, A. Hankin, J. P. Gaebler, D. Francois, A. Chernoguzov, D. Lucchetti, N. C. Brown, T. M. Gatterman, S. K. Halit, K. Gilmore, J. A. Gerber, B. Neyenhuis, D. Hayes, and R. P. Stutz, Realization of real-time fault-tolerant quantum error correction, Phys. Rev. X **11**, 041058 (2021).

[9] D. Bluvstein, S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kali-