ARTICLE OPEN

Check for updates

Better-than-classical Grover search via quantum error detection and suppression

Bibek Pokharel[1,2]✉ and Daniel A. Lidar[1,2,3,4]✉

We report better-than-classical success probabilities for a complete Grover quantum search algorithm on the largest scale demonstrated to date, of up to five qubits, using two different IBM platforms. This is enabled by error suppression via robust dynamical decoupling. Further improvements arise after the use of measurement error mitigation, but the latter is insufficient by itself for achieving better-than-classical performance. For two qubits, we demonstrate a 99.5% success probability via the use of the [[4, 2, 2]] quantum error-detection (QED) code. This constitutes a demonstration of quantum algorithmic breakeven via QED. Along the way, we introduce *algorithmic error tomography* (AET), a method that provides a holistic view of the errors accumulated throughout an entire quantum algorithm, filtered via the errors detected by the QED code used to encode the circuit. We demonstrate that AET provides a stringent test of an error model based on a combination of amplitude damping, dephasing, and depolarization.

npj Quantum Information (2024)10:23 ; https://doi.org/10.1038/s41534-023-00794-6

INTRODUCTION

The best possible classical strategy for finding a particular 'marked' element in an unsorted list of length N requires querying half of the elements in the list on average; a quantum computer (QC) can do this in quadratically fewer queries using Grover's search algorithm[1]. This algorithm is optimal and provably better than all classical strategies[2]. As one of the first algorithms with a provable quantum speedup, Grover search is often used as a subroutine for other quantum algorithms[3,4]. Over the last two decades, Grover search has been implemented on various quantum computing platforms[5–10], albeit for relatively small N.

Encoding a list of length N requires $n = \lceil \log_2(N) \rceil$ qubits. The list can be queried classically or using quantum queries; in both cases, one finds the marked element with some probability, which we refer to as the classical or quantum success probability. The largest implementation of Grover's algorithm to date is for $n = 8$ qubits, but without demonstrating a better-than-classical quantum success probability[5]. Such better-than-classical performance has been achieved for $n = 3$[6,7] and $n = 4$[8] qubits. Ref. [9] reported better-than-classical success probabilities for $n = 5$ for a single marked state, leaving open the possibility that the success probability would be reduced by averaging over all marked states. Here, employing two seven-qubit IBM Quantum Platform (IQP) transmon qubit platforms ibm_nairobi (Nairobi) and ibmq_jakarta (Jakarta), we demonstrate higher average success probabilities than all previous implementations, for $n \le 5$.

Key to our demonstrations is the use of error suppression and mitigation strategies. In particular, we use the [[4, 2, 2]] quantum error-detecting code[11,12], which encodes $k = 2$ logical qubits into $n = 4$ physical qubits and detects arbitrary single-qubit errors, to demonstrate a significant success probability enhancement relative to using two copies of $n = 2$ physical qubits. These success probabilities are further improved by combining error detection with measurement error mitigation[13,14].

We use the quantum error detection results to perform what we call *algorithmic error tomography*: for each encoded algorithm execution, based on the possible measurement outcomes we compute the probability of an error detectable by the [[4, 2, 2]] code or of a logical error. This allows us to compute a detailed map of the errors that arise at the conclusion of the entire algorithm. In this sense, algorithmic error tomography provides a holistic and complementary perspective to techniques such as gate set tomography[15,16], which instead focuses on individual gates applied during the algorithm.

We demonstrate better-than-classical performance for three or more physical qubits by employing error suppression via dynamical decoupling (DD)[17–20]. Toward this end, we consider three robust DD families: universally robust (UR)[21], concatenated DD (CDD)[22], and robust genetic algorithm (RGA)[23] sequences. We find that robust sequences with few pulses are vital in achieving better-than-classical algorithmic performance.

We compare the experimentally obtained results for Grover's algorithm with an error model based on the concatenation of amplitude damping, phase damping, and depolarization maps. Each map is parameterized by the calibration metrics provided by the IBM Quantum Platform (IQP) backend[24]. We test this model using the observed success probabilities and the algorithmic error tomography results; the latter provides a much more stringent test. We find good agreement with the model, but only after using DD. We interpret this in terms of the suppression of crosstalk by DD[25,26], which is unaccounted for by the error model.

In summary, we demonstrate a better-than-classical Grover search on up to 5 qubits, enabled by quantum error detection and dynamical decoupling. That is, we demonstrate algorithmic performance that is enhanced beyond the break-even point—where protected operations outperform their unprotected counterparts—and the capabilities of the best possible classical algorithm executing the same task. Along the way, we introduce algorithmic error tomography—a characterization of errors afflicting an entire quantum algorithm based on the syndromes of a quantum error detecting code.

[1]Department of Physics & Astronomy, Los Angeles, USA. [2]Center for Quantum Information Science & Technology, Los Angeles, USA. [3]Department of Electrical & Computer Engineering, Los Angeles, USA. [4]Department of Chemistry, University of Southern California, Los Angeles, CA 90089, USA. ✉email: pokharel@usc.edu; lidar@usc.edu

The structure of this paper is as follows. In the section "Grover's Algorithm: background and implementation", we summarize Grover's algorithm's salient aspects and discuss its implementation. In the section "Open system model", we describe the open system model we use to compute the theoretically expected algorithmic performance. Details about our dynamical decoupling implementation are in the section "Dynamical decoupling". The section "Two-qubit encoded Grover algorithm protected by quantum error detection" focuses on the performance of Grover's algorithm on $n = 2$ qubits with and without error detection. Algorithmic error tomography is introduced in the section "Two-qubit encoded Grover algorithm protected by quantum error detection" as well. The results for $2 < n \le 5$, where DD plays a crucial role in achieving better-than-classical performance, are given in the section "3-qubit to 5-qubit Grover's algorithm protected by dynamical decoupling". We conclude with observations and the implications of our results in the section "Discussion". Additional details and calculations in support of the main text are provided in the section "Methods".

RESULTS

Grover's Algorithm: background and implementation

Informally, the Grover problem is to search an unsorted list with $N = 2^n$ elements for a marked element. Formally, the goal is to find the marked n-bit bitstring m using the smallest number of queries of an oracle that implements a function $f_m: \{0,1\}^n \mapsto \{0,1\}$ defined as $f_m(x) = \delta_{x,m}$. Classically, after q queries, the probability of correctly identifying the marked element, which hereafter we refer to as the *success probability*, is $p_s^{\mathrm{C}}(q,N) = (q+1)/N$ (see the section "Classical success probability"). Consequently, the classical algorithm requires $O(N)$ queries.

Grover's algorithm provides a quadratic quantum speedup, requiring only $O(\sqrt{N})$ queries[1]. This scaling remains valid with more than one marked element[27], or even for an arbitrary initial amplitude distribution over the list elements[28]. In the original setting of a single marked element, the state after q queries to the oracle is

$$\left|\psi_q\right\rangle = \sin[(2q+1)\theta]|m\rangle + \cos[(2q+1)\theta]\left|m^{\perp}\right\rangle, \tag{1}$$

where $|m^{\perp}\rangle = \frac{1}{\sqrt{N-1}}\sum_{x\neq m}|x\rangle$ and $\theta = \arcsin\left(\frac{1}{\sqrt{N}}\right)$. Thus, the quantum success probability is $p_s^{\mathrm{Q}}(q,N) = \sin^2[(2q+1)\theta]$, and the theoretically optimal number of queries is $q_{\mathrm{opt}} = \lfloor \frac{\pi}{4}\sqrt{N} \rfloor$. Note that $p_s^{\mathrm{C}}(q,N) < p_s^{\mathrm{Q}}(q,N)$ for all $q < q_{\mathrm{opt}}$. However, the theoretically optimal q is often not experimentally optimal. As circuit depth increases with the number of queries and the problem size, there is a trade-off between the added decoherence and the increase in the success probability. Most experimental implementations of Grover's algorithm have focused on a single query[5–8], but this strategy does not scale well, as both $p_s^{\mathrm{C}}(1,N)$ and $p_s^{\mathrm{Q}}(1,N)$ decrease exponentially with n. We adopt an empirical approach to identify the optimal number of queries such that p_s is maximized. We set $q = 2$ for all problem sizes other than $n = 2$ where $q_{\mathrm{opt}} = 1$. We justify our choice of the number of queries in the section "Survey of dynamical decoupling sequences".

A schematic illustrating the implementation of the n-qubit Grover algorithm is shown in Fig. 1a. The only multi-qubit operation is the n-qubit controlled-phase gate $\mathrm{C}_{n-1}Z$, which needs to be implemented twice for each oracle query: once for the oracle and again for the amplitude amplification step. Different marked elements are represented by sandwiching the $\mathrm{C}_{n-1}Z$ gate with X_i or I_i depending on whether the corresponding bit b_i in the marked bitstring m is 0 or 1. I.e., letting $m = b_1 b_2 \ldots b_n$, then $\mathrm{C}_{n-1}Z$ in the oracle layer is preceded and followed by $X^{1-b_1} \otimes X^{1-b_2} \cdots \otimes X^{1-b_n}$. Likewise, amplitude amplification is implemented as $H^{\otimes n}X^{\otimes n}(\mathrm{C}_{n-1}Z)X^{\otimes n}H^{\otimes n}$.

For all problem sizes and oracles, we repeated each circuit for the maximum number of shots allowed on the QPU: 20,000 and 32,000 for Nairobi and Jakarta, respectively. The reported success probabilities were extracted by bootstrapping over these trials and all N possible marked states. All error bars reflect 95% confidence intervals obtained after bootstrapping unless specified otherwise.

Open system model

The QPUs used here are calibrated daily, and the following calibration metrics are recorded: the gate error e_{g} and gate duration τ_{g}, the qubit damping timescale T_1 and dephasing timescale T_2, and the response matrix M for readout errors (see Supplementary Information). In this section, we describe how we estimate the theoretical performance of Grover's algorithm using these metrics. The model described here is mathematically equivalent to the one used in Qiskit's Aer API's `NoiseModel.from_backend()` (see Supplementary information of Ref. [29]).

In a closed system described by a state ρ, a unitary gate U acts as $\mathcal{U}(\rho) = U\rho U^{\dagger}$. In reality, the system is open, so we model gate U as a CPTP map $\mathcal{E} = \mathcal{D} \circ \Phi \circ \mathcal{A} \circ \mathcal{U}$, where $\mathcal{D}, \mathcal{A}, \Phi$ are depolarizing, amplitude damping and phase damping maps respectively[30]. The amplitude damping and phase damping maps account for thermal relaxation, which we represent as $\mathcal{R} = \Phi \circ \mathcal{A}$. The single-qubit Kraus operators for $\mathcal{A} = \{A_0, A_1\}$ and $\Phi = \{F_0, F_1\}$ are

$$A_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-p_A} \end{pmatrix}, \quad A_1 = \sqrt{p_A}|0\rangle\langle 1| \tag{2}$$

$$F_0 = \sqrt{p_\Phi}I, \quad F_1 = \sqrt{1-p_\Phi}Z. \tag{3}$$

The n-qubit depolarizing map is

$$\mathcal{D} : \rho \mapsto (1-p_{\mathrm{D}})\rho + p_{\mathrm{D}}\frac{\mathbb{I}}{d} = \sum_j K_j \rho K_j^{\dagger}, \tag{4}$$

where $d = 2^n$ and $\mathcal{D}$ has d^2 Kraus operators:

$$K_0 = \sqrt{1 - \frac{d^2-1}{d^2}p_{\mathrm{D}}}\,\mathbb{I}^{\otimes n}, \tag{5}$$

$$K_j = \sqrt{\frac{p_{\mathrm{D}}}{d^2}}P_j, \quad P_j \in \{\mathbb{I}, X, Y, Z\}^{\otimes n} \backslash \mathbb{I}^{\otimes n}. \tag{6}$$

We parameterize these maps by their respective error probabilities p_{A}, p_Φ, and p_{D}, which in turn depend on the calibration metrics $e_{\mathrm{g}}, \tau_{\mathrm{g}}, T_1$, and T_2. In particular,

$$p_{\mathrm{A}} = 1 - \mathrm{e}^{-\tau_{\mathrm{g}}/T_1} \tag{7}$$

$$p_\Phi = \frac{1}{2}(1 + \mathrm{e}^{-\tau_{\mathrm{g}}/T_\Phi}) \tag{8}$$

$$p_{\mathrm{D}} = \frac{\mathrm{d}(F(\mathcal{R}) - 1 + e_{\mathrm{g}})}{\mathrm{d}F(\mathcal{R}) - 1}, \tag{9}$$

where

$$\frac{1}{T_\Phi} = \frac{1}{T_2} - \frac{1}{2T_1}, \tag{10}$$

and

$$F(\mathcal{E}) = \int \mathrm{d}\psi \langle\psi|U^{\dagger}\mathcal{E}(|\psi\rangle\langle\psi|)U|\psi\rangle = \frac{dF_{\mathrm{pro}}(\mathcal{E}) + 1}{d+1} \tag{11}$$

is the average gate fidelity for a CPTP map $\mathcal{E}$. Here $F_{\mathrm{pro}}(\mathcal{E})$ is the process fidelity of the map $\mathcal{E}$ with the target map $\mathcal{U}$, and d is the dimension of the map[31,32].

For each gate, we know the total gate error e_{g}, and so we compute p_{D} by setting $1 - e_{\mathrm{g}} = F(\mathcal{D} \circ \mathcal{R})$, which gives us Eq. (9)

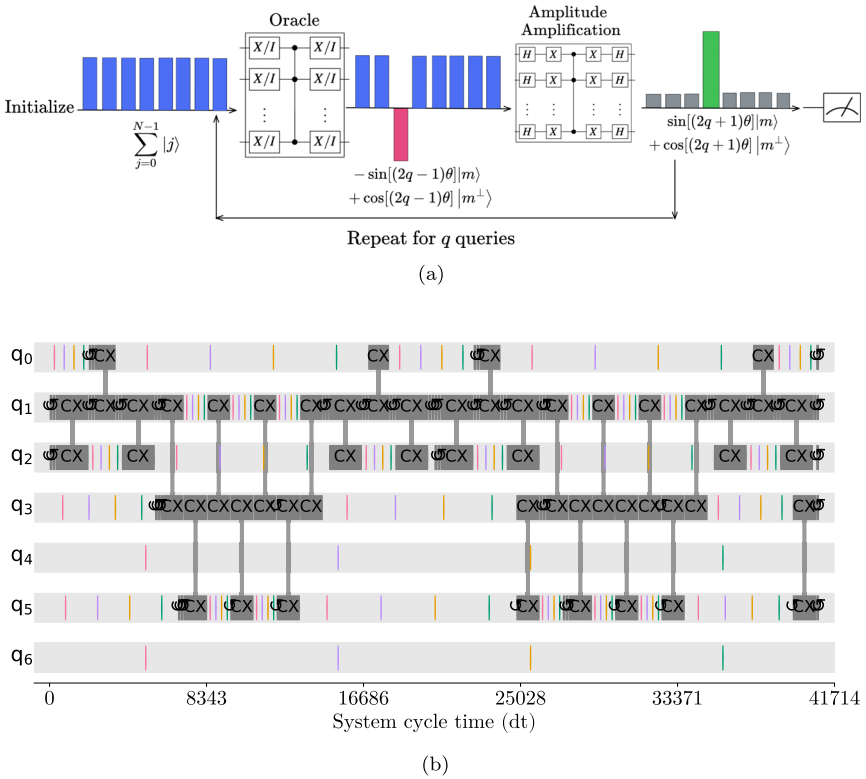


Fig. 1 Setup for Grover's algorithm and its implementation with DD. a Circuit description for Grover's algorithm. The relative amplitudes of all the states at each stage of the algorithm are shown. Starting with an equal superposition state, the oracle assigns a relative phase difference of π to the marked state. The amplitude amplification step then performs an inversion about the mean, allowing $|m\rangle$ to have a larger probability amplitude than all other states. This round of querying and amplifying is repeated q times. The optimal number of rounds for the n-qubit Grover problem is $q_{\text{opt}} = \lfloor \frac{\pi}{4} 2^{n/2} \rfloor$. The only multi-qubit operation required to implement both the oracle and the amplitude amplification step is $C_{n-1}Z$ (vertical line in the Oracle and Amplitude Amplification boxes). **b** Grover and DD. The timeline for one oracle query for 4-qubit Grover with the marked state $|1111\rangle$ is shown. Qubits q_4 and q_6 are spectators in this example. Recall that each oracle query for 4-qubit Grover requires two C_3Z gates. C_3Z requires 14 CNOTs, and the entire circuit uses 28 CNOTs; see the section "Circuit construction" for circuit compilation details. The pre-DD circuit elements are grayed out, and the colored lines represent the DD pulses. The DD sequence exemplified here uses four pulses for illustration purposes; in reality, we used longer sequences. The scheme demonstrated highlights four primary features of our implementation: (1) all idle intervals, including the ones on inactive qubits, are filled, (2) only one repetition of each sequence is performed, and the pulse interval is adjusted accordingly, (3) each pulse in the sequence can be unique, (4) a single qubit can experience multiple DD repetitions if there are multiple idle intervals.

(see the section "Extracting the depolarizing parameter from gate errors" for more details). In other words, we assign any gate error not accounted for by relaxation to depolarization.

We require that $p_{\text{D}} \geq 0$ (i.e. $F(\mathcal{R}) > 1 - e_{\text{g}}$) and if this condition is not met, then we assume that the error is entirely due to depolarization. In particular, when $F(\mathcal{R}) \leq 1 - e_{\text{g}}$, we set $p_{\text{D}} = e_{\text{g}} d/(d-1)$ and $\mathcal{E} = \mathcal{D} \circ \mathcal{U}$. For idle intervals in the circuit, no gate error e_{g} is reported by IQP[24], and therefore we model idle intervals with duration τ as identity operations where only the relaxation $\mathcal{R}$ matters. This is equivalent to setting the gate error for idle intervals to $e_{\text{idle}} = 1 - F(\mathcal{R})$.

In summary, we model single-qubit gates $U_{1\text{Q}}$ with gate duration τ_{g} as

$$\mathcal{D}(p_{\text{D}}) \circ \mathcal{R}(T_1, T_2, \tau_g) \circ \mathcal{U}_{1\text{Q}}, \tag{12}$$

and two-qubit gates $U_{2\text{Q}}$ with duration τ_{g} acting on qubits j, k are modeled as

$$\mathcal{D}(p_{\text{D}}) \circ [\mathcal{R}_j(T_1^j, T_2^j, \tau_{\text{g}}) \otimes \mathcal{R}_k(T_1^k, T_2^k, \tau_{\text{g}})] \circ \mathcal{U}_{2\text{Q}} \tag{13}$$

where $\mathcal{D}(p_{\text{D}})$ has 16 Kraus operators for $n = 2$ [see the section "Open system model"]. Idle intervals with duration τ are described by

$$\mathcal{R}(T_1, T_2, \tau). \tag{14}$$

Note that while we implement $\mathcal{D} \circ \mathcal{R}$, Qiskit's API reverses this order. While these channels do not commute, we found no significant difference between these two orderings in our simulations. In order to account for systematic measurement errors, the response matrix M is applied to all reported probabilities.

The quantum circuit for each experiment is first compiled into the QPU's native gate set and then scheduled using IQP's API[24]. We use this circuit to determine the order of operations and then replace each unitary map $\mathcal{U}$ with the corresponding CPTP channel $\mathcal{E}$. In the end, we acquire a probability distribution corresponding to the theoretical estimate of the circuit's output as measured in the computational basis.

Dynamical decoupling

DD is an open-loop quantum control technique wherein a sequence of pulses is strategically inserted between gates to suppress unwanted system-bath interactions[17–20]. While DD is fully compatible with quantum error correction[33], its most economical form requires no encoding, measurements, or post-processing. It is, therefore, perhaps the least resource-intensive error suppression strategy. Error suppression via DD has a long

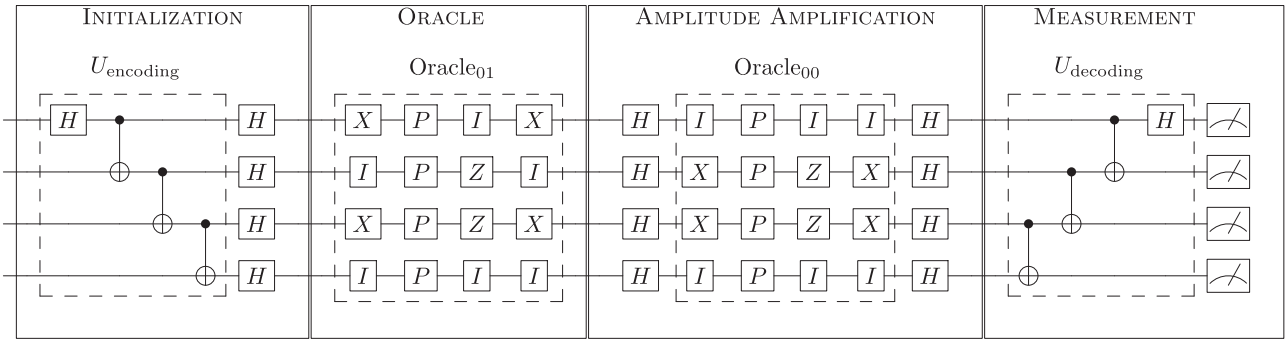


Fig. 2 Encoded two-qubit Grover. The two-qubit single-query ($q = 1$) Grover circuit encoded using the [[4, 2, 2]] code for the marked state $|01\rangle$ is shown. The encoding step (left dashed box) prepares the encoded initial state $|\overline{00}\rangle = \frac{1}{\sqrt{2}}(|0000\rangle + |1111\rangle)$ (the 4-qubit GHZ state) from the physical initial state $|0000\rangle$. The encoded Grover circuit is implemented by converting each physical gate of the $n = 2$ case of Fig. 1a into its logical counterpart, which is then converted into a physical 4-qubit implementation (middle left and right dashed boxes). The amplitude amplification step is the 00-oracle sandwiched by Hadamard gates. See the section "Circuit construction" for further details. $P = \text{diag}(1, i)$ is the phase gate. We post-select the measured results by decoding (right dashed box) and discarding any measurement outcome for which the code detects errors, i.e., does not result in one of the four decoded states $\{|0000\rangle, |0010\rangle, |0111\rangle, |0101\rangle\}$. Note that the circuit is not meant to be fault-tolerant, so decoding is acceptable.

history of experimental demonstrations on various quantum devices (see ref. [34] for a review). Here, we employ a 'decouple then compute' strategy[35,36], whereby control pulses constituting short but complete DD sequences are interleaved with the quantum circuit, exploiting intervals when individual qubits in the corresponding quantum circuits are idle. A scheme demonstrating our strategy is shown in Fig. 1b. This interleaving strategy has been used to improve quantum volume[37], variational quantum algorithms[38], and most recently to demonstrate an algorithmic quantum speedup[39].

In addition to the popular basic DD sequences—CPMG[40] and XY4[18]—we consider three robust sequence families: universally robust (UR) DD[21], concatenated DD (CDD)[22], and robust genetic algorithm (RGA) DD[23] (see the section "Survey of dynamical decoupling sequences" for more details). Other than CPMG, these are all high-order, multi-axis sequences that are universal for single qubits, i.e., they suppress arbitrary single-qubit errors beyond first order in the Magnus or Dyson expansion[41]. Robustness refers to the mitigation of axis-angle and over/under-rotation errors. In addition, these sequences can cancel crosstalk errors[25,26]. Our sequence choice is informed by the results of ref. [42], which reported on a significantly more comprehensive survey of sequences using superconducting qubits and concluded that robust sequences are preferred default choices. Here we do not utilize the OpenPulse functionality of the IQP platforms, nor do we implement Uhrig-type[43] non-uniform pulse interval DD sequences such as quadratic DD (QDD)[44], which were also found to perform well in the survey[42]. Both have the potential to enhance our results and are attractive options for future studies.

Two-qubit encoded Grover algorithm protected by quantum error detection

The [[4, 2, 2]] code[11] is the smallest possible qubit-based error-detecting code[12] and has been invoked for proof-of-principle demonstrations of quantum error detection[45]. Notably, it has been used to improve Clifford gate set fidelities[46] and the performance of variational algorithms[47]. However, measurement error mitigation (MEM) played a dominant role in Ref. [47], and it is unclear if error detection alone would have improved performance in that work. Here, we compare the performance of the two-qubit Grover algorithm with and without the [[4, 2, 2]] code and MEM. The unencoded version needs two qubits, while the encoded version requires four. To equalize resources, we simultaneously use two copies of the unencoded circuit and report the best fidelity of the two copies. We incorporate MEM using iterative Bayesian unfolding via the pyIBU package[48] (see the Supplementary Information for details about MEM). Ultimately, we demonstrate a conclusive improvement in algorithmic performance due to quantum error detection.

Encoding into the [[4, 2, 2]] code

The stabilizers of the [[4, 2, 2]] code are *XXXX* and *ZZZZ*. The logical operators of this code can be chosen as $\overline{X}_1 = XIXI, \overline{X}_2 = XXII, \overline{Z}_1 = ZZII$, and $\overline{Z}_2 = ZIZI$. Two-qubit Grover also requires the encoded Hadamard $\overline{H}$ and controlled-phase $\overline{CZ}$. The deconstruction of the logical circuit into physical components is detailed in the section "Circuit construction", and the resultant encoded two-qubit Grover circuit is shown in Fig. 2.

The encoding and decoding circuits, U_{enc} and $U_{\text{enc}}^{\dagger}$, are also depicted in Fig. 2. The logical basis states are:

$$|\overline{00}\rangle = U_{\text{enc}}|0000\rangle = \frac{|0000\rangle + |1111\rangle}{\sqrt{2}} \tag{15}$$

$$|\overline{01}\rangle = U_{\text{enc}}|0010\rangle = \frac{|0011\rangle + |1100\rangle}{\sqrt{2}} \tag{16}$$

$$|\overline{10}\rangle = U_{\text{enc}}|0111\rangle = \frac{|0101\rangle + |1010\rangle}{\sqrt{2}} \tag{17}$$

$$|\overline{11}\rangle = U_{\text{enc}}|0101\rangle = \frac{|1001\rangle + |0110\rangle}{\sqrt{2}} \tag{18}$$

These are also the four possible marked states in the two-qubit Grover problem. Consequently, after applying $U_{\text{enc}}^{\dagger}$ to decode the results, only states from the set $\mathcal{I} = \{|0000\rangle, |0010\rangle, |0111\rangle, |0101\rangle\}$ could have arisen from valid logical states. Therefore, we postselect by removing any of the 12 measurement outcomes that do not correspond to valid logical states, i.e., states in $\mathcal{I}^{\perp} = \{0, 1\}^4 \backslash \mathcal{I}$. This does not guarantee that all measurement outcomes in $\mathcal{I}$ correspond to the correct marked state: logical errors that prepare the wrong marked state are undetected by the code and also appear as measurement outcomes in $\mathcal{I}$ (more on this below). However, since we know which marked state the circuit was supposed to prepare, we can still reliably check whether a given valid measurement outcome (in $\mathcal{I}$) corresponds to the intended marked state.

Algorithmic error tomography

While the states in $\mathcal{I}^{\perp}$ do not correspond to valid (encoded) Grover algorithm outcomes, there is important information in such outcomes. They allow us to diagnose the frequency with which different sets of errors corresponding to the different syndromes of the [[4, 2, 2]] code appear at the end of the circuit (but before decoding) for a given encoded marked state. In this sense, these are the effective, or cumulative errors the code can detect at the conclusion of the algorithm, which is why we refer to the procedure we are about to describe as *algorithmic error tomography* (AET).

Let us first consider the different ways errors can occur before the measurement outcome is obtained. In principle, multiple errors of arbitrary weight can occur at multiple locations anywhere during the circuit, including between the decoding and measurement steps or even during the measurement. We will assume that the latter are perfectly implemented and show later that even decoding or measurement errors have a detectable signature. In other words, we assign a location to the cumulative errors as if they happened just before decoding, i.e., between the Amplitude Amplification and Measurement boxes in Fig. 2.

Then, given an $[[n, k, 2]]$ stabilizer error-detecting code $\mathcal{C}$, in AET, we treat all these errors either as detectable or as logical errors. Formally, if $|\overline{b}\rangle \in \mathcal{C}$ is a code basis state, where $b \in \{0,1\}^k$, E is an error, $\overline{U}_{\rm alg}$ is the unitary implementing the algorithm over the given encoding, $U_{\rm dec} = U_{\rm enc}^{\dagger}$ is the decoding unitary for $\mathcal{C}$, and $\mathcal{M}$ denotes a projective measurement in the computational basis, then in AET we interpret each measurement outcome $b' \in \{0,1\}^n$ as having arisen from $\mathcal{M}U_{\rm dec}E\overline{U}_{\rm alg}|\overline{0}^k\rangle$. More specifically, in the Grover case $\overline{U}_{\rm alg}|\overline{0}^k\rangle = |\overline{b}\rangle$, where $\overline{U}_{\rm alg}$ implements Grover's algorithm for the marked state b (the first three boxes in Fig. 2).

$\mathcal{M}$ denotes a simultaneous measurement of the Pauli observables $\{Z_i\}_{i=1}^n$, and each value of b' is one of the 2^n possible measurement outcomes. These outcomes label the 2^k logical outcomes of the logical Z operators $\{\overline{Z}_i\}_{i=1}^k$ and the 2^{n-k} error syndromes of the code $\mathcal{C}$. Each error syndrome corresponds to a particular set $\{(-1)^{s_i} \in \{-1,1\}\}_{i=1}^{n-k}$ of eigenvalues of the $n-k$ stabilizer generators $\{S_i\}_{i=1}^{n-k}$, which we denote by $s_i = 0$ or $s_i = 1$ for the eigenvalue 1 or -1, respectively. Thus, the no-error subspace is labeled by the bitstring $s = 0^{n-k}$, while all other bitstrings $s \in \{0,1\}^{n-k} \setminus 0^{n-k}$ label errors that are detectable by the code. The remaining k bits $\{z_i\}_{i=1}^k$ label the measurement outcomes of the logical Z operators, i.e., the eigenvalues $(-1)^{z_i}$ of $\overline{Z}_i$, with $1 \le i \le k$. Decoding the error syndromes and the logical outcomes can be done by backpropagating the individual Z_i observables through the decoding circuit. This is best illustrated via the example of the [[4, 2, 2]] code, as we do next.

In the case of the [[4, 2, 2]] code, the four measured bits correspond to the eigenvalues of the stabilizer generators $S_1 = XXXX$ and $S_2 = ZZZZ$ and two logical operators $\overline{Z}_1$ and $\overline{Z}_2$. To see this, we backpropagate the Z_i observables of each qubit through the $U_{\rm decoding}$ circuit in Fig. 2. Each time Z_i passes a gate, we conjugate it by the gate's inverse unitary (quantum evolution in the Heisenberg picture). For example, $Z_1 \mapsto HZ_1H = X_1 \mapsto \text{CNOT}\, X_1\, \text{CNOT} = X_1X_2 \mapsto \ldots \mapsto X_1X_2X_3X_4$, while $Z_2 \mapsto \text{CNOT}\, Z_2\, \text{CNOT} = Z_1Z_2 \mapsto \ldots \mapsto Z_1Z_2$, etc. In this manner, we find that given a measured bitstring $b' = b'_1b'_2b'_3b'_4$:

1. First (top) qubit: $(-1)^{b'_1} = XXXX$
2. Second qubit: $(-1)^{b'_2} = ZZII = \overline{Z}_1$
3. Third qubit: $(-1)^{b'_3} = IZZI = \overline{Z}_1\overline{Z}_2$
4. Fourth qubit: $(-1)^{b'_4} = IIZZ = ZZZZ\,\overline{Z}_1$,

Table 1. Mapping between measured bitstrings and syndromes as computed from Eq. (20).

Measured bitstring $b' = b'_1b'_2b'_3b'_4$	error syndrome $s = s_1s_2$
{0000, 0010, 0101, 0111}	00
{0100, 0110, 0011, 0001}	01
{1000, 1010, 1111, 1101}	10
{1100, 1110, 1011, 1001}	11

The four bitstrings in the first row correspond to the no-error syndrome, and we accept the corresponding values of the logical operators $\overline{Z}_1, \overline{Z}_2$. The remaining 12 bitstrings can be grouped into three error syndromes. These bitstrings correspond to measurement outcomes indicating that an error has been detected. We use the corresponding measurement results for AET.

where we have equated the operator with its eigenvalue in a slight abuse of notation. Consequently, $S_1 = XXXX = (-1)^{b'_1}$ and $S_2 = ZZZZ = (-1)^{b'_2+b'_4}$, so that the error syndrome is given by

$$s_1 = b'_1, \quad s_2 = b'_2 + b'_4. \tag{19}$$

The logical outcomes are given by $\overline{Z}_1 = (-1)^{b'_2}$ and $\overline{Z}_2 = (-1)^{b'_2+b'_3}$, i.e.,

$$z_1 = b'_2, \quad z_2 = b'_2 + b'_3. \tag{20}$$

It is simple to verify that the four states in the set $\mathcal{I}$ are then the unique set for which $s_1 = s_2 = 0$ (no error detected) while $\{z_1, z_2\} \in \{00, 01, 10, 11\}$. Each of the remaining 12 bitstrings corresponds to a non-trivial error syndrome; e.g., suppose that we measure the bitstring $b' = 1011$. Then $s_1 = 1, s_2 = 0 + 1 = 1$, so an error is detected. Note that this string also yields $z_1 = 0$ and $z_2 = 0 + 1 = 1$ as a logical measurement outcome, but this outcome must be ignored since it is accompanied by the detection of an error. The complete mapping from measured bitstrings b' to error syndromes s is given in Table 1.

As this analysis illustrates, the 4-qubit Hilbert space decomposes as $\mathcal{H} = \mathcal{C} \oplus \mathcal{C}^{\perp}$, where the code space $\mathcal{C} = U_{\rm enc}\,\text{span}\,(\mathcal{I})$ corresponds to the four measurement outcomes $s_1 = s_2 = 0$ with $z_1, z_2 \in \{00, 01, 10, 11\}$, and the error subspace $\mathcal{C}^{\perp} = U_{\rm enc}\,\text{span}\,(\mathcal{I}^{\perp})$ corresponds to the remaining 12 measurement outcomes where s_1 and s_2 are not both zero. We can further decompose $\mathcal{C}^{\perp}$ into the three syndrome subspaces $\mathcal{C}^{\perp}_{01} \oplus \mathcal{C}^{\perp}_{10} \oplus \mathcal{C}^{\perp}_{11}$, where the subscripts denote s_1 and s_2 (z_1 and z_2 are arbitrary).

However, there is more we can extract from the measurement outcomes than just the syndromes, as shown in Table 2. Namely, each bitstring b' arises from a particular set of Pauli errors we can straightforwardly identify. Consider, e.g., the Grover circuit for the marked state $|\overline{00}\rangle$. As mentioned above, in AET we treat the errors occurring during this circuit as an effective Heisenberg-picture Pauli error E between the conclusion of the circuit and the start of the decoding. Hence, $U_{\rm dec}E|\overline{00}\rangle$ represents the state arising prior to measurement, and when it is measured on a computational basis we obtain a bitstring b'. We can now exhaustively identify the complete set of Pauli errors E giving rise to the same b', starting from $|\overline{00}\rangle$, by solving $|b'\rangle = U_{\rm dec}E|\overline{00}\rangle$ for all compatible 4-qubit Pauli errors E (we are slightly abusing notation by writing $|b'\rangle$ in the binary representation instead of ± 1). This set is given in the bottom half of the *error outcome table* (Table 2), where each column is labeled by the corresponding measurement outcome b', and the first four rows correspond to the marked states $|\overline{00}\rangle, \ldots, |\overline{11}\rangle$. For example, the entry 0001

Table 2. Error outcome table.

Marked state	b'															
$\|\overline{00}\rangle$	0000	0010	0111	0101	0001	0011	0110	0100	1000	1010	1111	1101	1001	1011	1110	1100
$\|\overline{01}\rangle$	0010	0000	0101	0111	0011	0001	0100	0110	1010	1000	1101	1111	1011	1001	1100	1110
$\|\overline{10}\rangle$	0111	0101	0000	0010	0110	0100	0001	0011	1111	1101	1000	1010	1110	1100	1001	1011
$\|\overline{11}\rangle$	0101	0111	0010	0000	0100	0110	0011	0001	1101	1111	1010	1000	1100	1110	1011	1001
	$s=00$	$s=00$	$s=00$	$s=00$	$s=01$	$s=01$	$s=01$	$s=01$	$s=10$	$s=10$	$s=10$	$s=10$	$s=11$	$s=11$	$s=11$	$s=11$
Error type	$\boxed{IIII}$	IIXX	IXIX	IXXI	$\boxed{IIIX}$	$\boxed{IIXI}$	$\boxed{IXII}$	IXXX	$\boxed{IIIZ}$	*IIXY*	*IXIY*	*IXXZ*	$\boxed{IIIY}$	*IIXZ*	*IXIZ*	*IXXY*
	$\boxed{XXXX}$	*IIYY*	*IXZY*	*IXYZ*	*IIZY*	*IIYZ*	*IXZZ*	*IXYY*	$\boxed{IIZI}$	*IIYX*	*IXZX*	*IXYI*	*IIZX*	$\boxed{IIYI}$	*IXZI*	*IXYX*
	$\boxed{YYYY}$	*IZXY*	*IYIY*	*IYXZ*	*IZIY*	*IZXZ*	*IYIZ*	*IYXY*	$\boxed{IZII}$	*IZXX*	*IYIX*	*IYXI*	*IZIX*	*IZXI*	$\boxed{IYII}$	*IYXX*
	$\boxed{ZZZZ}$	*IZYX*	*IYZX*	*IYYI*	*IZZX*	*IZYI*	*IYZI*	*IYYX*	*IZZZ*	*IZYY*	*IYZY*	*IYYZ*	*IZZY*	*IZYZ*	*IYZZ*	*IYYY*
	IIZZ	*IXII*	*XIXI*	*XIIX*	*XXXI*	*XXIX*	*XIXX*	$\boxed{XIII}$	*XXXY*	*XXIZ*	*XIXZ*	*XIIY*	*XXXZ*	*XXIY*	*XIXY*	*XIIZ*
	IZIZ	*IXXZ*	*XIYZ*	*XIZY*	*XXYZ*	*XXZY*	*XIYY*	*XIZZ*	*XXYX*	*XXZI*	*XIYI*	*XIZX*	*XXYI*	*XXZX*	*XIYX*	*XIZI*
	IZZI	*XYIZ*	*XZXZ*	*XZIY*	*XYXZ*	*XYIY*	*XZXY*	*XZIZ*	*XYXX*	*XYII*	*XZXI*	*XZIX*	*XYXI*	*XYIX*	*XZXX*	*XZII*
	XXYY	*XYZI*	*XZYI*	*XZZX*	*XYYI*	*XYZX*	*XZYX*	*XZZI*	*XYYY*	*XYZZ*	*XZYZ*	*XZZY*	*XYYZ*	*XYZY*	*XZYY*	*XZZZ*
	XYXY	*YXIZ*	*YIXZ*	*YIIY*	*YXXZ*	*YXIY*	*YIXY*	*YIIZ*	*YXXX*	*YXII*	*YIXI*	*YIIX*	*YXXI*	*YXIX*	*YIXX*	$\boxed{YIII}$
	XYYX	*YXZI*	*YIYI*	*YIZX*	*YXYI*	*YXZX*	*YIYX*	*YIZI*	*YXYY*	*YXZZ*	*YIYZ*	*YIZY*	*YXYZ*	*YXZY*	*YIYY*	*YIZZ*
	YXXY	*YYII*	*YZXI*	*YZIX*	*YYXI*	*YYIX*	*YZXX*	*YZII*	*YYXY*	*YYIZ*	*YZXZ*	*YZIY*	*YYXZ*	*YYIY*	*YZXY*	*YZIZ*
	YXYX	*YYZZ*	*YZYZ*	*YZZY*	*YYYZ*	*YYZY*	*YZYY*	*YZZZ*	*YYYX*	*YYZI*	*YZYI*	*YZZX*	*YYYI*	*YYZX*	*YZYX*	*YZZI*
	YYXX	*ZIXY*	*ZXIY*	*ZXXZ*	*ZIIY*	*ZIXZ*	*ZXIZ*	*ZXXY*	$\boxed{ZIII}$	*ZIXX*	*ZXIX*	*ZXXI*	*ZIIX*	*ZIXI*	*ZXII*	*ZXXX*
	ZIIZ	*ZIYX*	*ZXZX*	*ZXYI*	*ZIZX*	*ZIYI*	*ZXZI*	*ZXYX*	*ZIZZ*	*ZIYY*	*ZXZY*	*ZXYZ*	*ZIZY*	*ZIYZ*	*ZXZZ*	*ZXYY*
	ZIZI	*ZZXX*	*ZYIX*	*ZYXI*	*ZZIX*	*ZZXI*	*ZYII*	*ZYXX*	*ZZIZ*	*ZZXY*	*ZYIY*	*ZYXZ*	*ZZIY*	*ZZXZ*	*ZYIZ*	*ZYXY*
	ZZII	*ZZYY*	*ZYZY*	*ZYYZ*	*ZZZY*	*ZZYZ*	*ZYZZ*	*ZYYY*	*ZZZI*	*ZZYX*	*ZYZX*	*ZYYI*	*ZZZX*	*ZZYI*	*ZYZI*	*ZYYX*

The top half gives the possible measurement outcomes b' for each of the four (encoded) marked states prepared by the Grover oracle. The bottom half gives the Pauli error types that could have generated the measurement outcome in the same column, for a given marked state. In the fifth row, we list the corresponding syndrome, computed using Eq. (20). The boxed operators in the first column are the four stabilizers of the [[4, 2, 2]] code. Every other operator corresponding to the $s = 00$ syndrome (the first four columns) is a logical error or logical operator. The other boxed operators highlight all the weight-1 errors, of which 4/12 belong to a single column of the $s = 10$ syndrome. Columns 5–12 correspond to the detectable errors. Errors are listed in lexicographical order in each column.

in column 5 of the row marked by $|\overline{00}\rangle$ means that the measurement outcome was $b' = 0001$ for the oracle marking $|\overline{00}\rangle$, and the complete set of errors that could have given rise to this measurement outcome is *IIIX*, *IIZY*, *IZIY*, … (see column 5). The [[4, 2, 2]] code does not distinguish between these errors but does detect them.

We re-emphasize that these errors are in the Heisenberg picture, i.e., they are the result of forward-propagating physical errors from the start of the circuit to the location corresponding to the point between the end of the Amplitude Amplification step and the start of the Measurement step in Fig. 2 (indeed, we derived them by back-propagating each Z_i from the end of the Measurement step). Therefore they do not necessarily represent physical errors at this location, and an error operator such as *ZIII* listed in column 9 of Table 2 should not be confused with a physical *Z* error on the first qubit, since this *ZIII* error could have arisen from an actual physical *XIII* error if the latter took place between the *P* and *H* gates in the Amplitude Amplification step (see Fig. 2). Nevertheless since all of the gates in the encoded two-qubit Grover circuit after the CNOT gates of the Initialization step are single-qubit gates, any weight-1 physical error appearing after these CNOT gates will remain a weight-1 error in the Heisenberg picture. This means that if weight-1 physical errors dominate the higher-weight errors, then this will manifest as a predominance of a single column of the $s = 10$ syndrome, since by inspection of Table 2, column 9 corresponds to the largest number of weight-1 errors: four out of twelve. We will see the manifestation of this remark in the experimental results below.

The first four columns in Table 2 correspond to undetectable errors: the errors they list commute with the stabilizers of the [[4, 2, 2]] code (boxed in the first column) and correspond to logical errors. For example, $IXYZ|\overline{00}\rangle = i|\overline{11}\rangle$. Thus, even if the measurement yields a bitstring in $\mathcal{I}$ (the first four columns), the state prepared by the circuit could have been the wrong marked state. Of course, this is only problematic when we do not know the correct answer in advance; in the present case, since we program the circuit, we can immediately check whether a given measurement outcome in $\mathcal{I}$ is the correct answer or one of the three other valid logical states.

We can now construct the *algorithmic error tomography (AET) table*: given a marked state *b*, each observed state $|b'\rangle$ will have an empirical probability $p_{b'|b} = N_{b'}/N_{\text{tot}}$, where $N_{b'}$ is the number of times $|b'\rangle$ is observed out of a total of N_{tot} observations. The AET table thus gives us the empirical probabilities associated with the different errors classified in the error outcome table (Table 2).

Figures 3 and 4 show the AET tables after implementing or simulating the encoded two-qubit Grover algorithm on Jakarta and Nairobi, respectively. Each of the four panels corresponds to a different AET table, with the top row representing experiments and the bottom row representing simulations using the model of the section "Open system model". The left and right panels of Fig. 3 exclude or include DD, respectively. We did not use DD in the Nairobi case (Fig. 4). Each row corresponds to a different encoded marked state $z = z_1 z_2$ within each syndrome table. Other than the logical errors counted during postselection (first column), all other columns represent outcomes ignored during the postselection step. More precisely, the columns represent the empirical probabilities of observing the bitstrings starting from column 5 of Table 2.

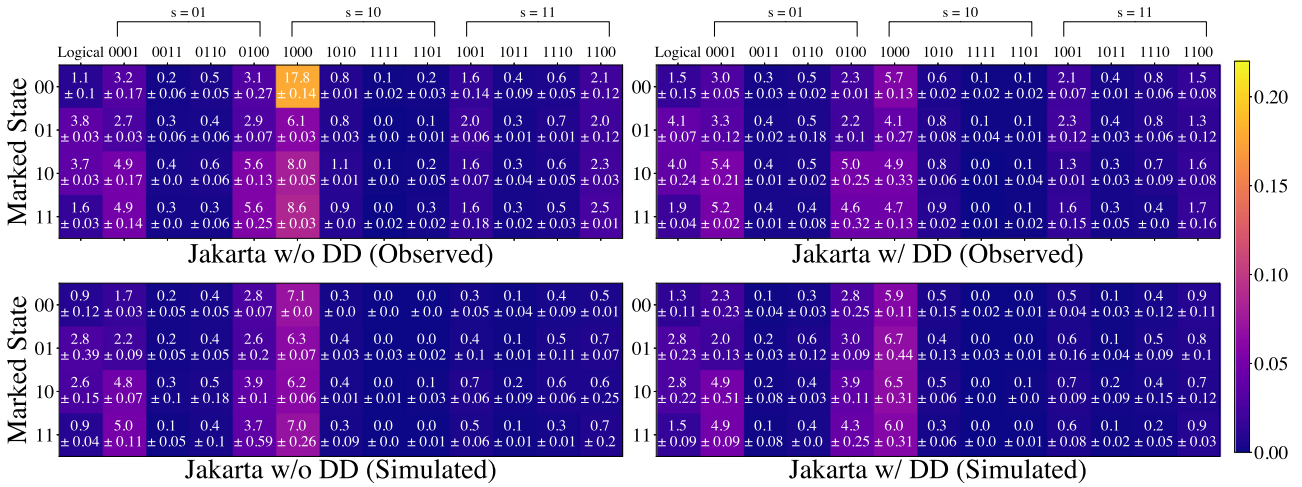


Fig. 3 AET table for Jakarta. The bitstring observed after U_{dec} either corresponds to a marked entry or an error tabulated in Table 2. Column headers correspond to the bitstrings for the marked state $|\overline{00}\rangle$, but the values inside the table are organized identically to Table 2 (starting from its first row, column 5). I.e., the c'th column of each of the four tables shown corresponds to the $c+4$'th column of Table 2. For example, the percentages in the second row and fifth column of each of the four tables shown correspond to measuring the bitstring $b' = 1010$ when the marked state is $|\overline{01}\rangle$ (since this is the bitstring found in column 9 of Table 2 in the row of $|\overline{01}\rangle$). The complete set of errors that could have given rise to this outcome is listed in column 9 of Table 2. Top: experimental results. The numbers in each box are the empirical percentage probabilities with 2σ standard deviation. Logical error percentage probabilities are shown in the first column of each table. Each row corresponds to a different marked state. The probabilities in each row do not sum to unity since we do not display the probability of obtaining the correct marked state. Left: without DD protection. Right: with DD protection. Bottom: the same for the simulated model.

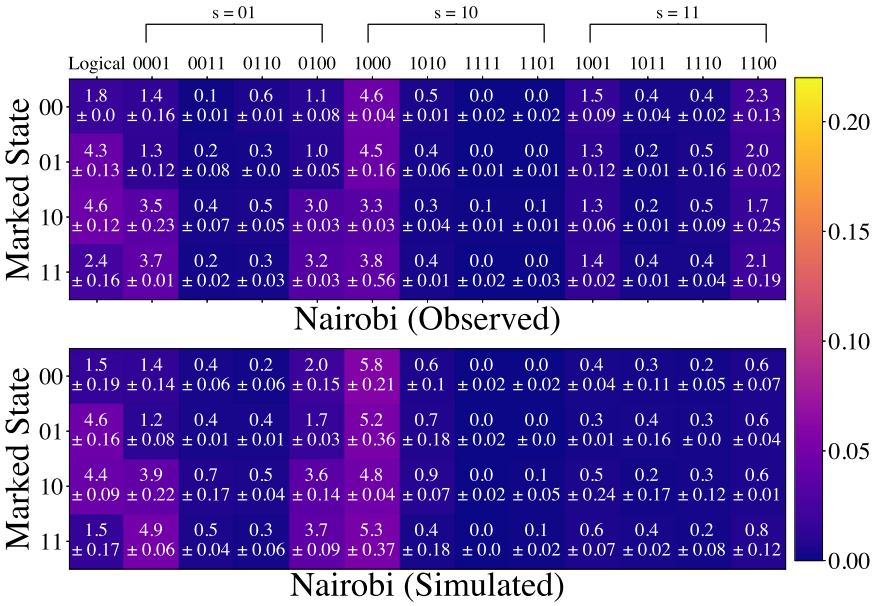


Fig. 4 AET table for Nairobi. Data entries are as in Fig. 3, except that only data without DD is shown. Good agreement is observed between the results of our error model and the experiment. In particular, compared to the AET table for Jakarta (Fig. 3), we do not observe an asymmetry in $s = 10$ type errors across marked states.

The column headers of the AET tables are the first row (i.e., marked state $|\overline{00}\rangle$) of Table 2.

It is visually clear from both Figs. 3 and 4 that the percentages in column 5 are higher than the rest. This result can be explained in terms of the above analysis of errors in the Heisenberg picture. Namely, column 5 corresponds to the errors in column 9 of Table 2, which contains the largest number of weight-1 errors. The data presented in Figs. 3 and 4 thus support the notion that the error processes underlying weight-1 errors are dominant. This analysis highlights the utility of AET as a diagnostic of the dominant errors in a given circuit and suggests that error-correcting codes can be correspondingly tailored and optimized.

We discuss these results in more detail in the following section, showing how AET in addition allows us to identify and mitigate qubit crosstalk.

DD protection and comparison with the open system model

Recall that we define the success probability p_s as the probability of correctly identifying the marked element. We denote the empirical

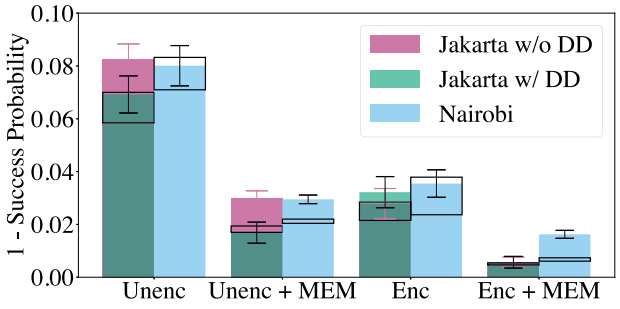


Fig. 5 Two-qubit Grover results. Two-qubit single-query Grover failure probability results without (Unenc) and with (Enc) postselection using the $[[4,2,2]]$ code on Jakarta and Nairobi are shown. The transparent boxes represent the theoretically expected failure probabilities from the model described in the section "Open system model", which does not include DD; their centers correspond to the average over marked states, and their boundaries correspond to 95% confidence intervals after bootstrapping. The colored bars represent the experimental results (see the legend), and the experimental error bars (black for Jakarta with DD and Nairobi, or pink for Jakarta without DD) correspond to 95% confidence intervals after bootstrapping. Dark green appears where the pink and light green colors (i.e., Jakarta with and without DD) overlap. In the Unenc case, we run two identical copies of the two-qubit Grover problem to equalize resources with the Enc case and choose the copy with the highest success probability. Also shown are the results with MEM using iterative Bayesian unfolding (see the Supplementary Information for details). Failure probabilities with and without DD protection are shown for Jakarta but not for Nairobi, where the simulated and observed error tomography and failure probabilities are in agreement (see the section "DD protection and comparison with the open system model"). The presence of DD does not affect the success probability in the encoded implementation, and as a result, the pink bars are mostly hidden behind the green bars. However, the nature of detected errors, even in the encoded case, is affected by DD (see Fig. 3). All data for different runs on the same QPU were collected on the same day; data from different QPUs were collected on different days.

success probability obtained for a list of N elements after q oracle queries by $p_s^e(q,N)$. Our results are summarized in Fig. 5, which shows the failure probability $(1-p_s^e(1,4))$ for the unencoded and the encoded implementations on two different QPUs.

Before comparing the results with and without encoding, we analyze whether the observed performance matches the model of the section "Open system model" in both cases. Let us focus first on Jakarta, where without DD, the empirical failure probabilities in the unencoded case are slightly higher than predicted; see the leftmost column of Fig. 5. Fortunately, in the encoded case, Jakarta's failure probability overlaps with the prediction bands (Fig. 5, third column from the left). However, a closer look at the detected errors via AET reveals a different discrepancy. The simulated results for Jakarta (bottom-left panel in Fig. 3) do not quite match the empirical error profile (top-left panel of Fig. 3), in particular in the 1000 column for the marked state $|\overline{00}\rangle$ (the yellow entry), which captures the errors listed in column 9 of Table 2. In other words, Jakarta does not quite match the simulations for unencoded or encoded circuits without DD: the empirical results show stronger syndrome $s=10$-type errors and also a stronger marked state-dependent asymmetry in these errors than the simulations. In contrast, for Nairobi, Fig. 4 shows that the AET simulation results essentially agree with the empirically observed ones. This also holds for the simulated failure probabilities (Fig. 5).

To investigate Jakarta's observed discrepancy, we first attempt to systematically amplify p_D, T_1, and T_2 by multiplying each quantity by a phenomenologically determined variable λ_i (see Supplementary Information). This leads to a better overlap between predicted and observed success probabilities but does not reproduce the AET asymmetry seen in Fig. 3. This shows the limitations of the phenomenological model of the section "Open system model" and highlights the level of detail provided by AET.

However, the Jakarta discrepancy is effectively removed after the application of DD, as can be seen by comparing the top-right and bottom-right panels of Fig. 3. Our two-qubit Grover implementation uses four qubits, leaving three inactive qubits in the 7-qubit QPUs used in our experiments. As there are no idle intervals in the unencoded two-qubit Grover circuit, we applied the XY4 sequence on the inactive qubits—q_2,q_4, and q_6 (see Supplementary Information). We applied the XY4 sequence to both the active and inactive qubits for the encoded case. Due to the relative sparsity of idle intervals in the two-qubit Grover circuits, we did not attempt to implement robust sequences, which require more pulses than XY4.

Figure 5 shows how the failure probability and the rates of various detected errors in Jakarta are affected by the presence of DD. For the unencoded case (the first two columns from the left of Fig. 5), DD improves the performance slightly, and the discrepancy between the predicted and observed failure probabilities is removed. The improvement by DD in the unencoded two-qubit Grover case is in concurrence with refs. [25,26], which showed the efficacy of the XY4 sequence in suppressing static ZZ crosstalk in superconducting qubits. In other words, these results confirm that ZZ crosstalk—which is well-documented for superconducting QCs[49]—likely contributes to the observed performance being slightly worse than expected from the model.

Adding the XY4 sequence removes most of the empirical-theoretical discrepancies in both the magnitude and the asymmetry of the errors exhibited by the AET profiles, as seen by comparing the top and bottom right of Fig. 3. With DD, the encoded circuits have a weaker state-wise asymmetry in $s=10$-type errors than seen in the left column of Fig. 3. Moreover, the DD-protected circuits more closely reproduce the distribution of detected errors predicted by the model of the section "Open system model" than the same circuits without DD. This observation—that the agreement between our theoretical model and the experimental results improves under DD—is further validated below.

The close agreement we found for Nairobi between our (crosstalk-free) model and the experimental results without DD or MEM (Fig. 4 and the first and third from left columns of Fig. 5) suggests that crosstalk does not play a significant role in this QPU. Figure 5 does exhibit a significant discrepancy between the model and the experimental Nairobi results when MEM is included (second and last columns of Fig. 5). As we show in the Supplementary Information, this discrepancy arises from the choice to mitigate readout errors using iterative Bayesian unfolding (IBU)[14].

Finally, Fig. 6 complements the first and last columns of Fig. 5, as well as the AET results, and shows the output distributions for Jakarta and Nairobi for the two-qubit Grover case, with and without encoding and MEM. The main observation is that for the unencoded case, the maximum success probability is obtained for the marked state $|00\rangle$, which is also the QPU's ground state; this is unsurprising given the dominance of amplitude damping errors. With encoding plus error mitigation, the overall performance increases and becomes independent of the marked state.

Success probability: beyond break-even improvement

We now focus on the effect of error detection on two-qubit Grover performance as seen in Fig. 5. Due to the shallow circuit depth, even without any error detection, $p_s^e(1,4)\sim 93.0\%$—already much higher than the classical success probability $p_s^C(1,4)=\frac{1}{2}$. Adding error detection improves the success

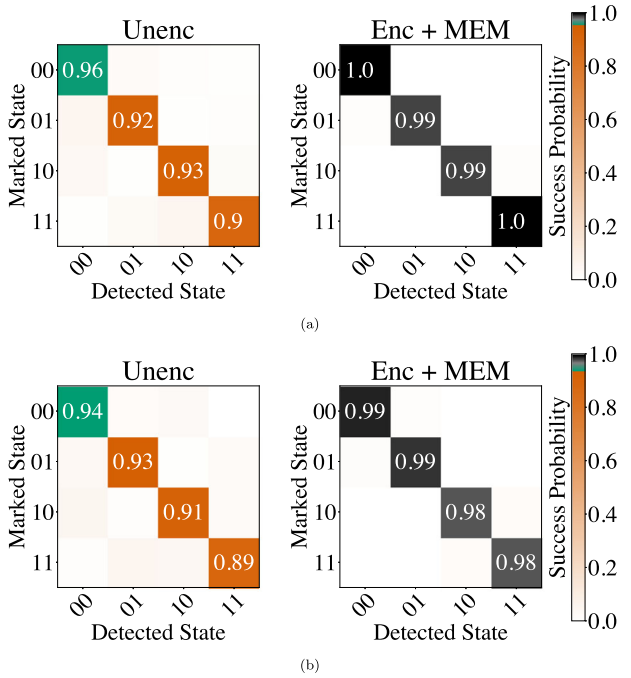


Fig. 6 Two-qubit Grover results. The left and right panels show the output distribution for Jakarta (**a**) and Nairobi (**b**) with all possible oracles and two setups: unencoded and encoded with MEM. As in Fig. 5, Unenc corresponds to two copies of unencoded two-qubit Grover, of which the best result is reported. Enc corresponds to the results encoded using the $[[4, 2, 2]]$ code. The Enc results are reported after postselection. Let $p_s(m, b, e)$ be the observed success probability for marked state m, detected state b, and experiment type $e \in \{\text{Enc}, \text{Unenc} + \text{MEM}, \text{Enc} + \text{MEM}\}$. The success probability changes from orange to green when $p_s(m, b, e) > \max_{m,b} p_s(m, b, \text{Unenc})$. Error bars correspond to 95% confidence intervals.

probability to ~96.0%. The effect of MEM is similar to that of error detection: the success probability increases to ~97.0%. Combining error detection with MEM results in additional improvement: we obtain success probabilities of ~98.5% on Nairobi and ~99.5% on Jakarta. Due to error detection and MEM, Jakarta's success probabilities increase by an order of magnitude.

This improvement over the unencoded case is non-trivial, considering that the $[[4, 2, 2]]$ code can only detect weight-1 errors, and the encoded circuit requires six two-qubit gates. In contrast, the unencoded version requires only two. The relatively high success probabilities we observe in the encoded case suggest that most errors, even those due to the two-qubit gates, manifest as weight-1 errors. This shows, albeit for a relatively small problem size, that error detection can more than offset the extra errors introduced due to increased circuit depth and complexity.

We have demonstrated an algorithmic beyond-break-even improvement using error detection in the sense that the protected algorithm clearly outperforms its unprotected counterpart. Previous break-even improvements were at the individual gate level[50,51]. Here we have demonstrated such an improvement at the level of the execution of an entire algorithm, albeit of a fixed size. The holy grail is to demonstrate the implementation of an algorithm for a family of problem sizes at the logical level with higher fidelity than the same algorithm executed at the physical level. Achieving this in our setting would require increasing the problem and code sizes. The family of $[[2k + 2, k, 2]]$ subsystem quantum error detecting codes is an attractive option in this regard since all their logical operators can be chosen to be 2-local[52], which simplifies the circuit design. An experimental implementation of such larger codes and problem sizes remains a coveted goal.

3-qubit to 5-qubit Grover's algorithm protected by dynamical decoupling

Crossing the classical threshold in Grover's search for an increasingly larger number of qubits is a meaningful goal, not only because the quadratic speedup offered by Grover's algorithm leads to a more dramatic improvement as the problem size increases but also because it becomes more challenging to realize the speedup experimentally as the controlled phase gate $C_{n-1}Z$ is an n-qubit entangling operation. In the implementation of ref. [5], 5-qubit Grover required nearly a thousand two-qubit gates, and for 8-qubit Grover, nearly 15,000 gates were used. Notably, this exponential increase in the number of two-qubit gates with problem size is because ref. [5] did not use ancilla qubits to make the circuits shallower. It is possible to implement $C_{n-1}Z$ with circuits where two-qubit gates scale linearly with n (see the section "Circuit construction"). Ref. [8] employed shallower circuits for $C_{n-1}Z$ and solved 5-qubit Grover with slightly better-than-random success probabilities but without better-than-classical performance. Ref. [9], which improved upon ref. [8], has reported the highest Grover's algorithm success probabilities to date for $n = 5$, using a Honeywell ion-trap device; however, only a single marked state ($|01011\rangle$) was tested. Whether the success probability averaged over all 2^5 marked states crosses the classical threshold remains to be tested. Ref. [8]'s results on superconducting-qubit-based QCs for the same marked state had lower success probabilities than reported in our experiments. Using quantum multiprogramming, ref. [10] improved algorithmic success probabilities for quantum partial search (a variation of the canonical Grover search) even on superconducting-qubit-based devices. However, quantum multiprogramming could not increase algorithmic performance for the canonical Grover algorithm. While noteworthy, Ref. [10]'s approach requires modifying Grover's algorithm and substantially increasing the number of physical qubits used in the algorithm. Here our goal is to provide a comprehensive demonstration of better-than-classical performance for Grover's algorithm, along with a theoretical model for the open-system effects that explain our results.

We use an efficient, ancilla-assisted implementation of generalized Toffoli-type gates[8,53] to implement $C_{n-1}Z$ (see the section "Circuit construction"). Our implementation, which builds upon the circuits from ref. [8], uses 8, 14, and 22 CNOTs for a single $C_{n-1}Z$ gate for $n = 3$, 4, and 5, respectively. The deepest circuit we implement is two oracle queries for 5-qubit Grover, totaling 88 CNOTs. Despite being far shallower than ref. [5]'s implementation, this is still a relatively deep circuit; e.g., the quantum supremacy demonstration of ref. [54] and the algorithmic quantum speedup demonstration of ref. [39] involved circuits of depth up to 40 and 44, respectively. As we detail next, owing to error suppression via DD, we crossed the classical probability threshold for all problem sizes, including for 5-qubit Grover.

Implementation with dynamical decoupling

DD sequences are inserted into idle intervals of a quantum circuit using the 'decouple then compute strategy' demonstrated in Fig. 1b, which shows the DD insertion scheme for a single query on a 4-qubit Grover circuit. In contrast to two-qubit Grover, where we restricted DD implementation to XY4, $n \geq 3$ has ample idle intervals. Therefore we can implement robust sequences (RGA, CDD, UR) requiring more than four pulses. Each of these families has multiple members that are parameterized by the number of pulses in the sequence. We restrict our implementation to DD sequences with fewer than 32 pulses and, for each circuit, only consider sequences that we can fit in the idle intervals available in the quantum circuit.

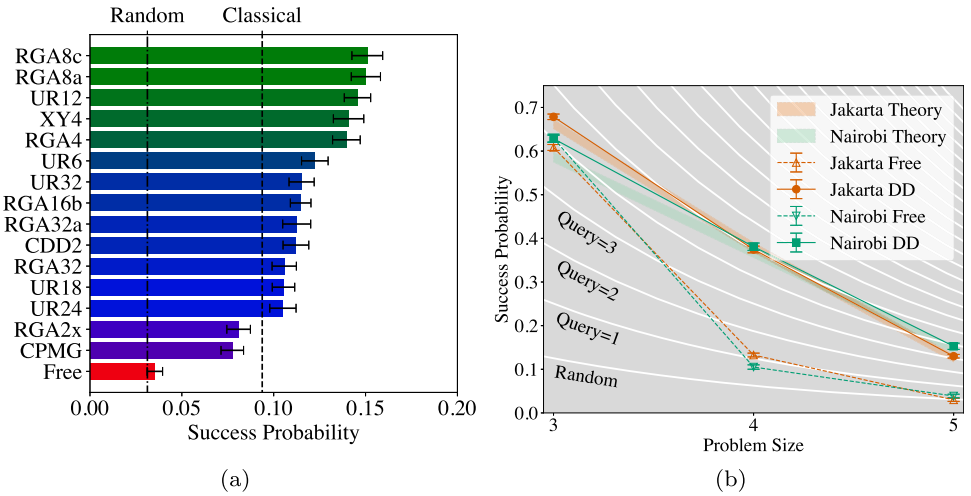


Fig. 7 Performance of Grover with DD. a Average success probability for 5-qubit Grover with two oracle queries on Nairobi. The DD sequences are ranked in order of decreasing success probability. The two dotted lines represent success probabilities corresponding to a random and classical strategy, respectively. RGA8a and RGA8c are tied as the best-performing sequences. Free denotes the result of an unprotected implementation. Error bars correspond to 99% confidence intervals. **b** Success probabilities *vs* problem size. Nairobi (green) and Jakarta (orange) success probabilities for $n \in \{3, 4, 5\}$ are shown for DD-protected and unprotected implementations. The translucent bands indicate the theoretically estimated success probabilities using the model described in the section "Open system model". We performed $q = 2$ queries to the quantum oracle in all cases. The ideal success probabilities are 0.945, 0.908, and 0.602 for $n = 3$, 4, and 5, respectively. The white lines correspond to the success probabilities for the classical strategy and random sampling from the unsorted list ($q = 0$). Error bars correspond to 99% confidence intervals.

At each problem size n, there are 2^n possible oracles, each corresponding to one marked state $|b\rangle$, where $b \in \{0,1\}^n$. We proceed as follows to avoid implementing this exponentially large set of oracles. Given $0 \le k \le n$, there are $\binom{n}{k}$ distinct bitstrings that are identical to 0^k1^{n-k} up to qubit permutation. Recall that marked states differ only by whether X or I gates surround the $C_{n-1}Z$ gate. Thus, we only consider the $n+1$ oracles with marked states $|0^k1^{n-k}\rangle, k \in \{0, \ldots, n\}$. We then estimate the average success probability by computing

$$\langle p(n)\rangle = \frac{1}{2^n}\sum_{k=0}^{n}\binom{n}{k} p\left(|0^k1^{n-k}\rangle\right). \tag{21}$$

We use $\langle p(n)\rangle$ as the metric for selecting the optimal DD sequences among those we tested and to identify the experimentally optimal number of queries $q^{\mathrm{e}}_{\mathrm{opt}}$. Once the optimal DD sequence and $q^{\mathrm{e}}_{\mathrm{opt}}$ are identified for each n, we run the unprotected and the DD-protected Grover's algorithm again at $q^{\mathrm{e}}_{\mathrm{opt}}$, but this time for all 2^n oracles.

Optimal DD sequence and number of queries

Our first goal is to identify the best DD sequence and q_{opt}. The determinations made in this step inform our choices for the next step. For conciseness, in this section, we focus on the results of the largest problem size we implemented, i.e., $n = 5$. The section "Survey of dynamical decoupling sequences" shows the results for $3 \le n \le 5$ on both Nairobi and Jakarta. The performance of various DD sequences for Nairobi for 5-qubit Grover are compared in Fig. 7a by computing $\langle p(5)\rangle$. The unprotected evolution (Free) is marginally better than choosing an element randomly and does not cross the classical threshold. DD protection is necessary to cross this threshold, but the two-pulse sequences RGA2x and CPMG still result in worse-than-classical performance. The RGA and UR sequences perform well, particularly those with fewer than 12 pulses. RGA8a and RGA8c are tied as the best-performing sequences; we choose RGA8a for the next step, where we implement all of the 2^5 oracles. The performance improvement seen due to robust sequences is consistent across problem sizes and devices, as detailed in the section "Survey of dynamical decoupling sequences".

We also use $\langle p(n)\rangle$ to identify the experimentally optimal number of oracle queries $q^{\mathrm{e}}_{\mathrm{opt}}$. The theoretically optimal number of repetitions for $n = 3, 4, 5$ is $q_{\mathrm{opt}} = 2, 3, 4$, respectively. However, the section "Survey of dynamical decoupling sequences" shows that in reality, the theoretically expected q_{opt} often leads to worse performance than $q^{\mathrm{e}}_{\mathrm{opt}}$. For the DD-protected implementation, $q = 2$ maximizes the success probability p_{s} in all cases other than 5-qubit Grover on Jakarta, where the performance at $q = 2$ is comparable to $q = 1$. As DD protection is necessary to cross the classical threshold, for simplicity of analysis and to maximize p_{s} we set $q = 2$ from here on.

Better-than-classical performance

Figure 8 shows our results for the 5-qubit Grover problem on Nairobi with and without DD. Even in our relatively shallow-depth implementation, before error suppression via DD (whether Free or Free + MEM), the final results are indistinguishable from randomly guessing the marked state. The results change significantly when we implement DD. With DD, the classical threshold is crossed by all marked states. Adding MEM improves the results slightly, but only when accompanied by DD.

This dramatic improvement due to DD holds for other problem sizes as well. Figure 7b shows the success probabilities after two oracle queries on both devices for $3 \le n \le 5$ (see Supplementary Information for the role of postselection in these results). At the two smaller problem sizes ($n = 3, 4$), the unprotected implementations are better than random sampling, but the success probability is relatively low. For $n = 4$, the unprotected quantum Grover circuit does not exceed the classical single-query threshold. It is effectively on par with random sampling for $n = 5$. In contrast, for all problem sizes, the DD-protected quantum strategy at $q = 2$ outperforms the

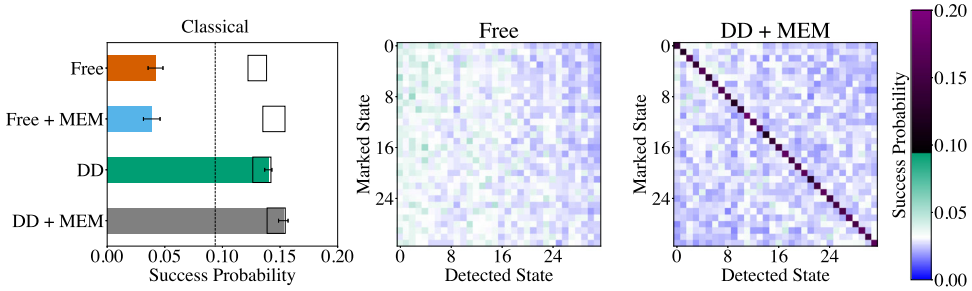


Fig. 8 5-qubit Grover results on Nairobi. Left: average success probability with and without DD or MEM for 5-qubit Grover implemented on Nairobi. The boxes correspond to the theoretically expected success probabilities. The quantum oracle is queried twice; in the ideal case, the success probability is 0.602. The unprotected (Free) evolution is on par with a random guess, significantly worse than the optimal classical strategy (dashed vertical line), and just adding MEM does not change the result. In contrast, the DD-assisted implementation crosses the classical threshold, and the results improve even more with MEM, up to a success probability of 0.15. Error bars correspond to 99% confidence intervals. Middle and right: the complete input-output maps for all 2^5 marked states, without and with DD + MEM, are shown. States are sorted by increasing Hamming weight; in the Free case, low Hamming weight states have a higher success probability (more green on the left). This is likely to be a consequence of amplitude damping (spontaneous emission), which favors the $|0\rangle$ state of each qubit. In the unprotected case (Free, middle), there is no discernible correlation between the input marked state and the output detected state. In the protected case (DD + MEM right), black-to-purple signifies better-than-classical success probability, and this threshold is crossed for all 32 marked states. The DD sequence used here is RGA8a[23], which was the top-performing sequence in our DD survey [see Fig. 7a].

classical strategy for $q \leq 3$. DD-protected Grover performance at $n = 3, 4, 5$ is equivalent to classical $q = 4, 5, 3$ for Jakarta and $q = 4, 5, 4$ for Nairobi, respectively. Thus, DD is essential in attaining a better-than-classical performance.

The translucent bands in Fig. 7b and the boxes in Fig. 8 (left) show the theoretically expected results computed from the open system model with the IQP-supplied parameters. The success probability in this unprotected Grover case [dashed lines in Fig. 7b] is considerably lower than the theoretical expectation. This discrepancy is likely due to crosstalk and non-Markovianity, which are well-documented for IQP's superconducting qubit-based QPUs. Once we use DD, the observed fidelities improve and are close to the theoretical predictions. This improvement is expected given DD's ability to reduce the effect of crosstalk[25,26] and non-Markovian effects.

With DD, the algorithmic performance approaches the expectations based on our error model. However, we emphasize that this model does not predict the QPU's performance under DD; it simply tells us what the performance would be if the reported calibration metrics corresponded to observed dynamics. The overlap between the theoretically predicted (translucent) and the DD-protected (solid) performance implies that DD successfully mitigates the errors that our simple model does not account for. However, the model does not provide an upper bound on the possible performance improvement due to error suppression. For instance, better-optimized sequences could suppress idle-time errors further, and dynamically corrected gates can suppress errors during operations[55,56].

We note that the restriction to $n \leq 5$ arose not because of circuit width but depth. In particular, we used two oracle queries, though theoretically $q_{\text{opt}} = 4$ at $n = 5$. The gap between the theoretically and experimentally optimal number of queries is expected to grow with problem size. As is true for any quantum algorithm, optimizing circuit compilation and increasing metrics such as T_1, T_2, and gate fidelities are all vital for scalability.

DISCUSSION

We implemented Grover's algorithm of various sizes on multiple superconducting qubit devices. To our knowledge, this is the largest successful demonstration of Grover's algorithm for which the quantum strategy outperforms its classical counterpart. For two-qubit Grover, we focused on error detection via the [[4, 2, 2]] code and showed that it allowed us to achieve near-optimal performance. Along the way, we introduced the method of algorithmic error tomography. We showed that it provides a wealth of information complementary to previous protocols, such as gate set tomography or just measuring the success probability of an algorithm. We showed that error suppression via DD is essential in attaining better-than-classical performance for larger problem sizes.

Grover's algorithm is a demanding algorithm[5] as it requires multiple implementations of $C_{n-1}Z$ – a fully entangling operation. The superconducting trimon device[6], which prior to our results achieved the highest success probability for 3-qubit Grover, is an example of algorithm-tailored hardware where C_2Z is a native gate. Constructing hardware that can natively perform such entangling operations may be one path to realizing the full potential of Grover's algorithm. Still, it is desirable to achieve this goal with more general-purpose quantum hardware, as we have strived to do here.

Today's quantum experimentalists have various error mitigation tools at their disposal. Measurement error mitigation[14,57], dynamical decoupling, zero noise extrapolation[13], and quantum error detection[41] are complementary strategies that address different kinds of errors. Whether and which error mitigation method to employ must be decided based on the problem and available resources. In this work, we combined MEM with DD and quantum error detection. As expected, these strategies complement each other. However, we found that often, MEM only became useful after DD was employed. Dynamical decoupling, which arguably has the lowest resource overhead and requires no postprocessing, was the single most effective strategy in improving the performance of our implementation of Grover's algorithm. Our work adds to the growing literature[25,37–39,42,58] on the effectiveness of error suppression through DD.

While we demonstrated a crossing of the classical threshold at every problem size we tested, better-than-classical success probabilities are not enough to claim a provable quantum speedup[59]. Such a claim would require computing the scaling of the time-to-solution metric (including the overhead of post-selection and error mitigation, if applicable) as a function of

problem size and extending it to the largest possible problem size that can be embedded on the device. Here we could not go to the largest possible problem size as even at $n = 5$, our circuit is quite deep—two queries required 88 two-qubit gates, and for a larger number of queries or qubits, we no longer observed a quantum advantage. Achieving quantum speedup for Grover search will require devices that can implement circuits much deeper than those used here without a catastrophic drop in fidelity. Recent results[60,61] suggest that without significant improvements in the implementation of the surface code, the latter will not necessarily provide an advantage in the context of Grover's algorithm (or other algorithms providing a quadratic speedup) over the type of error suppression and mitigation methods we have explored here. Thus, our results are likely to be necessary (but not sufficient) stepping stones toward a quantum speedup for Grover's algorithm.

METHODS

Classical success probability

Let $p_s^C(q,N)$ be the classical success probability after q oracle queries for an unsorted list with N elements. Suppose that we have made q calls to the oracle. The probability that the marked element is among the q queried elements is q/N. Since the oracle will confirm that the marked element is among these queries, the probability of correctly identifying the marked element is 1 in this case. Conversely, the probability that the marked element is among the remaining $N-q$ elements is $(N-q)/N$, and in this case the probability of correctly guessing the identity of the marked element is $1/(N-q)$. By the law of total probability, we then have $p_s^C(q,N) = \frac{q}{N} \times 1 + \frac{N-q}{N} \times \frac{1}{N-q} = \frac{q+1}{N}$.

Survey of dynamical decoupling sequences

In addition to the well-known XY4 and CPMG sequences, we consider three families of robust dynamical decoupling sequences. These sequences are expected to work well on a superconducting device with finite pulse width and flip-angle errors. The first sequence family is concatenated DD (CDD). CDD comprises recursively generated sequences by concatenating a base sequence such as the XY4 sequence. Formally,

$$\mathrm{XY4/CDD}_1 \equiv Y - X - Y - X \tag{22}$$

$$\mathrm{CDD}_n \equiv \mathrm{XY4}([\mathrm{CDD}_{n-1}]) \tag{23}$$

Here we could only proceed as far as CDD_2, as the idle intervals in the circuit were too short to incorporate higher-order CDD sequences.

The second family comprises the robust genetic algorithm (RGA) sequences[23]. These were found by assuming a generic single-qubit error term and a numerical optimization using genetic algorithms. A subset of the sequences was enforced to be robust against flip-angle errors. Therefore, these sequences are called robust genetic algorithm sequences. Due to duration constraints, we only attempted sequences up to 32 pulses, even though longer sequences were identified in ref. [23].

$$\mathrm{RGA}_4 \equiv \overline{Y} - X - \overline{Y} - X \tag{24}$$

$$\mathrm{RGA}_{4p} \equiv \overline{Y} - \overline{X} - \overline{Y} - \overline{X} \tag{25}$$

$$\mathrm{RGA}_{8a} \equiv X - \overline{Y} - X - \overline{Y} - Y - \overline{X} - Y - \overline{X} \tag{26}$$

$$\mathrm{RGA}_{8c}/\mathrm{EDD} \equiv X - Y - X - Y - Y - X - Y - X \tag{27}$$

$$\mathrm{RGA}_{16b} \equiv \mathrm{RGA}_{4p}([\mathrm{RGA}_{4p}]) \tag{28}$$

$$\mathrm{RGA}_{32a} \equiv \mathrm{RGA}_4([\mathrm{RGA}_{8a}]) \tag{29}$$

$$\mathrm{RGA}_{32c} \equiv \mathrm{RGA}_{8c}([\mathrm{RGA}_4]). \tag{30}$$

Here X means a π-rotation about the $+x$ axis. In contrast, $\overline{X}$ means a π-rotation about the—x-axis (see ref. [42] for a concise and detailed summary with more explicit definitions, including the effect of pulse width and the associated errors).

Finally, the third family is that of universally robust (UR) sequences[21]. UR sequences are defined such that

$$\mathrm{UR}_n = (\pi)_{\phi_1} - (\pi)_{\phi_2} - \ldots - (\pi)_{\phi_n} \tag{31}$$

$$\phi_k = \frac{(k-1)(k-2)}{2}\Phi^{(n)} + (k-1)\phi_2 \tag{32}$$

$$\Phi^{(4m)} = \frac{\pi}{m}\Phi^{(4m+2)} = \frac{2m\pi}{2m+1}, \tag{33}$$

where $(\pi)_\phi$ is rotation about the axis at an angle of ϕ from the $+x$-axis. We choose $\phi_1 = 0$, and $\phi_2 = \Phi(n)$ so that all UR_n sequences are palindromic. Once again, we constrained our survey to sequences with up to 32 pulses.

Our results from testing these three robust families of DD sequences are shown in Fig. 9. For $n > 3$, almost all DD sequences improved the success probability, but even among the sequences tried, there was considerable variation. Robust sequences with fewer than 12 pulses per DD cycle were the best performers. The eventual decrease in the performance of sequences with an increasing number of pulses is to be expected as they are implemented using noisy gates, and there is a trade-off between the protection provided by DD and the accumulation of gate errors. The RGA8c and RGA8a sequences performed consistently well and are the only sequences to cross the classical threshold on Jakarta for $n = 5$. RGA8c is also commonly known as the Eulerian DD (EDD) sequence, and RGA8a is a slightly modified version of EDD. These palindromic sequences are known to be robust against flip-angle and finite-width errors. The best sequence at each problem size is shown in Table 3.

Figure 9g and h shows the experimental success probabilities for the unprotected and DD-protected Grover circuits for all queries q. Here, only the best DD sequence from the survey above (listed in Table 3) is used in each case. Theoretically, for $n = 3, 4, 5$, $q_{\mathrm{opt}} = 2, 3, 4$ respectively. Unfortunately, for 5-qubit Grover, software restrictions prevented us from going beyond $q = 2$ and 3 on Jakarta and Nairobi, respectively. However, it is already clear that the experimentally optimal value was reached in both cases. Recall that we restricted our results to $q = 2$ oracle queries in the main text. For $n = 3$, this is both experimentally and theoretically optimal. For Nairobi, two queries have the highest experimental success probability for all problem sizes. For Jakarta and $n = 5$, a single query has a slightly higher success probability, but the difference between $q = 1$ and $q = 2$ is not substantial.

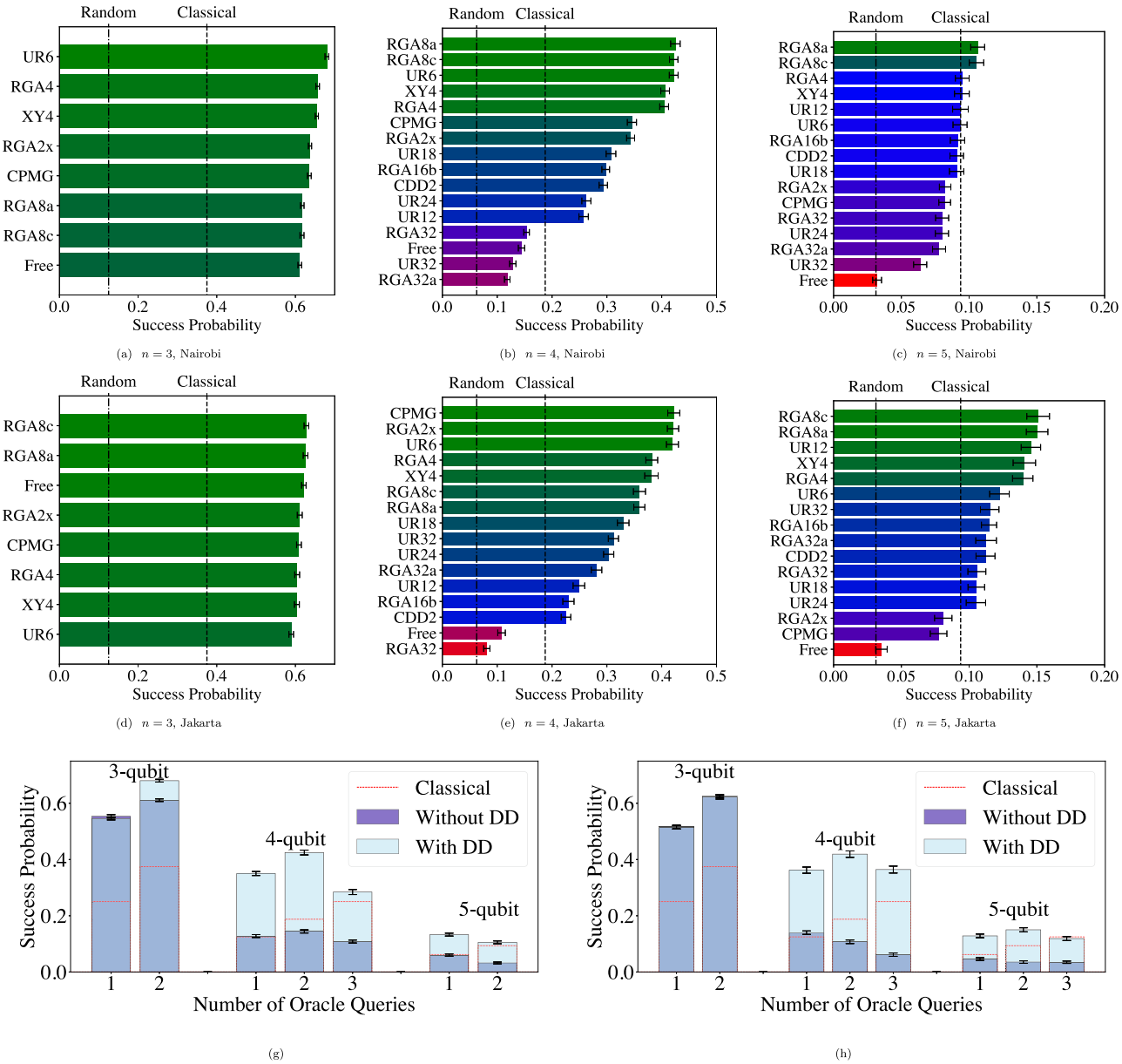


Fig. 9 Performance of DD sequences and query numbers. a–f Performance of DD sequences, expanding on the results shown in Fig. 7a. Average success probability for $n = 3, 4, 5$ with two oracle queries on Jakarta (**a**–**c**) and Nairobi (**d**–**f**). The DD sequences are ranked in order of decreasing success probability. The two dotted lines represent success probabilities corresponding to a random and classical strategy, respectively. For $n > 3$, the unprotected evolution (Free) is marginally better than choosing an element randomly and does not cross the classical threshold. DD protection is necessary to cross the classical threshold, and the RGA and UR sequences with fewer than 12 pulses are the best performers. Error bars correspond to 99% confidence intervals. **g** and **h** Performance under different oracle query numbers. Success probabilities are shown as a function of the number of oracle queries for Jakarta (left) and Nairobi (right). All results included MEM and error bars represent 99% confidence intervals. Dashed red lines correspond to the optimal classical success probability. Except for $n = 3$, the classical threshold is crossed only with DD. In the main text, we set $q = 2$, which is the optimal number of repetitions for all instances other than $n = 5$ on Jakarta. Error bars correspond to 99% confidence intervals.

Overall, for simplicity of analysis, in the main text, we focused only on results for $q = 2$.

Finally, Fig. 10 shows the results for all oracles at two queries using the DD sequence found from the survey above. The results are qualitatively identical on both devices. We have already clarified that DD is necessary to cross the classical threshold. One might suspect that majority voting may suffice to declare a detected state as the marked state if it is the mode of its corresponding probability distribution. However, even under this criterion, for 5-qubit Grover, there is no way to detect the marked state without DD.

Table 3. The best-performing DD sequence at each problem size for both QPUs.

Problem size	Jakarta	Nairobi
$n = 3$	UR6	RGA8a
$n = 4$	RGA8a	UR6
$n = 5$	RGA8c	RGA8a

These sequences were determined by implementing $n + 1$ oracles of the form $0^k 1^{n-k}$ for the n-qubit Grover problem.

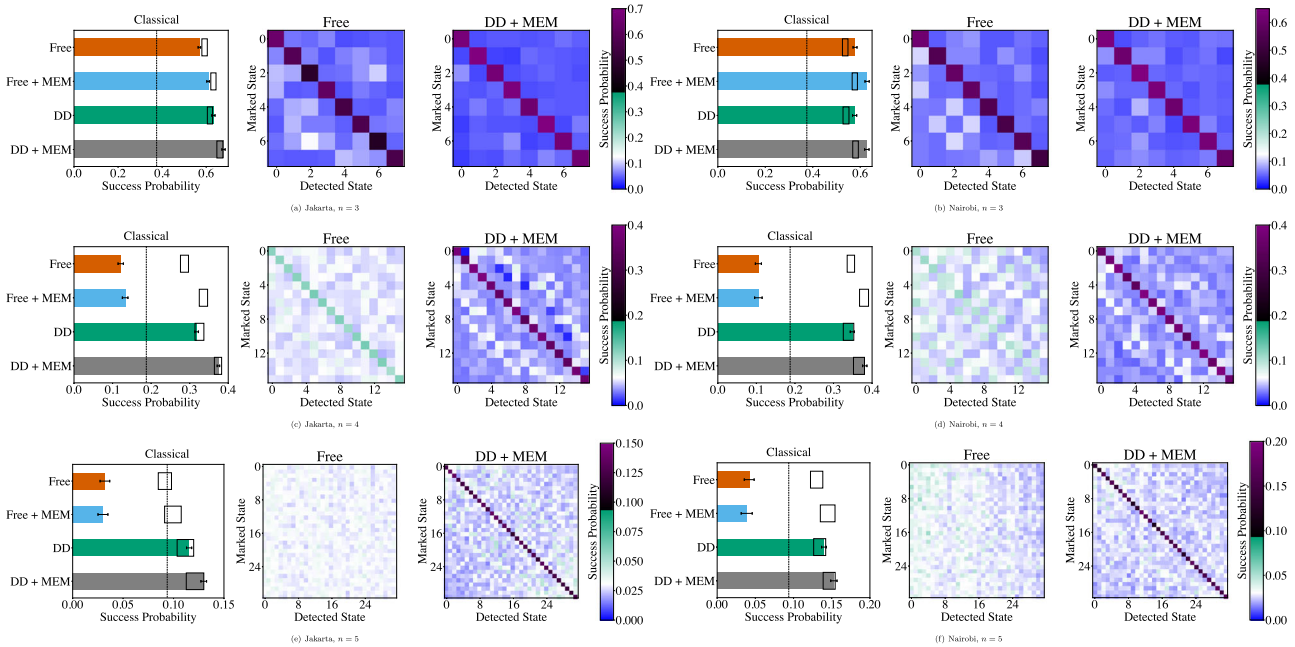


Fig. 10 Output distribution for 3-qubit, 4-qubit, and 5-qubit Grover. The panels show the output distribution for Jakarta (left) and Nairobi (right) after two oracle queries, complementing Fig. 8, which only shows Nairobi for $n = 5$. Each row represents a problem size in ascending order. In a row, the horizontal bar plot on the left shows the success probability under no error suppression and mitigation (Free), with measurement error mitigation (Free + MEM), with DD protection (DD), and with DD protection and measurement error mitigation (DD + MEM). The dashed horizontal line and the boxes represent the classical and the theoretically expected success probability, respectively. The second and third columns show the input-output map for Free and DD + MEM, highlighting the improvement offered by these strategies. The states are sorted by increasing Hamming's weight. The transition from green to black occurs at the classical success probability threshold. With DD protection, the classical threshold is crossed in all cases.

Extracting the depolarizing parameter from gate errors

IQP devices are calibrated daily, and for each gate, the gate error e_g and gate time τ_g are reported. The associated T_1 and T_2 times are also reported for each qubit. As e_g is extracted in the presence of thermal relaxation errors, $\mathcal{R} = \Phi \circ \mathcal{A}, p_D = f(e_g, \tau_g, T_1, T_2)$. In order to extract p_D from e_g, we assume that

$$e_g = 1 - F(\mathcal{D} \circ \mathcal{R}). \tag{34}$$

Using

$$F(\mathcal{D} \circ \mathcal{R}) = (1 - p_D)F(\mathcal{R}) + p_D F(\mathbb{D} \circ R), \tag{35}$$

where $\mathbb{D}$ is the completely depolarizing channel, we get

$$1 - e_g = F(\mathcal{D} \circ \mathcal{R}) \tag{36}$$

$$= (1 - p_D)F(\mathcal{R}) + p_D F(\mathbb{D} \circ R) \tag{37}$$

$$= (1 - p_D)F(\mathcal{R}) + p_D F(\mathbb{D}) \tag{38}$$

$$= (1 - p_D)F(\mathcal{R}) + \frac{p_D}{d}. \tag{39}$$

Consequently,

$$p_D = \mathrm{d}\frac{F(\mathcal{R}) - 1 + e_g}{\mathrm{d}F(\mathcal{R}) - 1}. \tag{40}$$

If we assume that there are no relaxation errors and only depolarizing noise affects the gate error e_g, then $p_D = \frac{d}{d-1} e_g$.

Circuit construction

Recall that implementing the n-qubit Grover's algorithm requires the $C_{n-1}Z$ gate, the only multi-qubit gate necessary for both the oracle and the amplitude amplification step. We provide circuit diagrams for how each n-qubit controlled phase gate, $C_{n-1}Z$, was transpiled. We rely on previously known circuit designs for our circuit construction, particularly the circuits used by ref. [8].

Two-qubit Grover circuits. For two-qubit Grover, CZ does not require transpilation as $CZ = H \cdot CX \cdot H$. However, a few nuances must be considered when constructing the encoded two-qubit Grover circuits. There are three components to a Grover circuit with marked element $m = b_1 b_2$: initialization into the state $|\psi\rangle$, oracle query O_m, and amplitude amplification A. More precisely,

$$|\psi\rangle = \frac{1}{4}\sum_{b_i, b_j = \{0,1\}} |b_j\rangle \otimes |b_k\rangle = H \otimes H|00\rangle \tag{41}$$

$$O_m = (X^{1-b_1} \otimes X^{1-b_2}) \cdot CZ \cdot (X^{1-b_1} \otimes X^{1-b_2}) \tag{42}$$

$$A = H \otimes H \cdot O_{00} \cdot H \otimes H. \tag{43}$$

To convert these circuits into their logical counterparts, we note that $\overline{X}_1 = XIXI, \overline{X}_2 = XXII, \overline{Z}_1 = ZZII, \overline{Z}_2 = ZIZI, H^{\otimes 4} = \overline{\mathrm{SWAP}}_{12}(\overline{H}_1\overline{H}_2)$ and $P^{\otimes 4} = (\overline{Z}_1\overline{Z}_2)\overline{CZ}$. It is also helpful to notice that $[\mathrm{SWAP}_{12}, U \otimes U] = 0$ for any unitary U and $[\mathrm{SWAP}_{12}, CZ] = 0$. Moreover, $\mathrm{SWAP}_{12}|b_1 b_1\rangle = |b_1 b_1\rangle$ and $[\mathrm{SWAP}_{12}, O_{00}] = 0$ as O_{00} only has operators of the form $U \otimes U$ and CZ. Consequently,

$$|\overline{\psi}\rangle = \overline{H} \otimes \overline{H}|\overline{00}\rangle \tag{44}$$

$$= \overline{H} \otimes \overline{H} \cdot \overline{\mathrm{SWAP}_{12}}|\overline{00}\rangle \tag{45}$$

$$= \overline{\mathrm{SWAP}_{12}} \cdot \overline{H} \otimes \overline{H}|\overline{00}\rangle \tag{46}$$

$$= H^{\otimes 4} U_{\mathrm{enc}}|0000\rangle. \tag{47}$$

Implementing the encoded Grover oracle is straightforward and

does not involve any two-qubit operations:

$$\overline{O_m} = \overline{X^{1-b_1}} \otimes \overline{X^{1-b_2}} \cdot IZZI \cdot P^{\otimes 4} \cdot \overline{X^{1-b_1}} \otimes \overline{X^{1-b_2}}. \tag{48}$$

Lastly,

$$A = \overline{H} \otimes \overline{H} \cdot \overline{O_{00}} \cdot \overline{H} \otimes \overline{H} \tag{49}$$

$$= \overline{H} \otimes \overline{H} \cdot \overline{O_{00}} \cdot \overline{H} \otimes \overline{H} \cdot \overline{\mathrm{SWAP}_{12}} \cdot \overline{\mathrm{SWAP}_{12}} \tag{50}$$

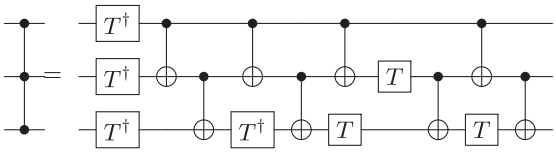
$$= \overline{\mathrm{SWAP}_{12}} \cdot \overline{H} \otimes \overline{H} \cdot \overline{O_{00}} \cdot \overline{\mathrm{SWAP}_{12}} \cdot \overline{H} \otimes \overline{H} \tag{51}$$

$$= H^{\otimes 4} \cdot \overline{O_{00}} \cdot H^{\otimes 4} \tag{52}$$

$$= H^{\otimes 4} \cdot IXXI \cdot IZZI \cdot P^{\otimes 4} \cdot IXXI \cdot H^{\otimes 4}. \tag{53}$$

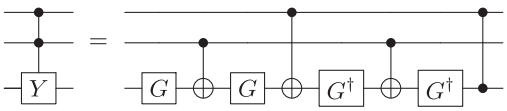
The corresponding circuits for the marked state $|01\rangle$ are shown in Fig. 2.

3-qubit to 5-qubit Grover circuits. The problem of transpilation increases in complexity with problem size. $C_{n-1}Z$ can be achieved by finding a circuit decomposition for the n-qubit Toffoli gate $C_{n-1}X$. It is known that the three-qubit Toffoli gate, C_2X, can be implemented using six CNOTs[62]. However, this requires a fully connected architecture. As no fully connected group of three qubits can be found in the QPUs we used, we rely on the 8-CNOT decomposition[8] of C_2Z shown in Eq. (54).



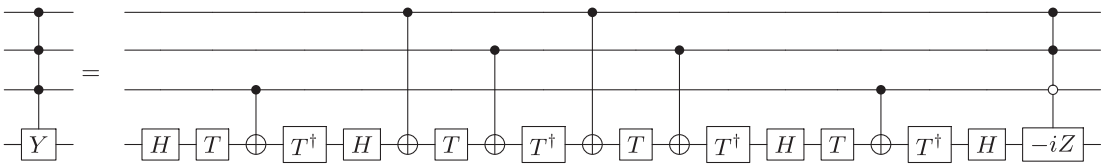
(54)

Here $T = Z^{1/4}$. For C_3Z and C_4Z we use relative-phase Toffoli gates[53]. Breaking down C_kZ using C_aZ and C_bY such that $a + b = k + c$ allows for C_kZ to be implemented with fewer CNOTs as long as we use c ancillas. In our construction, we only use one ancilla for C_3Z and C_4Z. C_2Y is shown in Eq. (55),



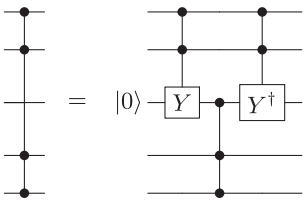
(55)

where $G = R_y(\pi/4)$, and C_3Y is shown in Eq. (56):



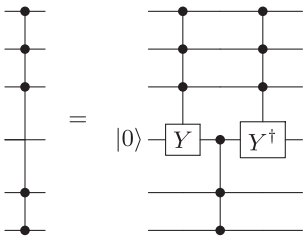
(56)

Finally, using the relative-phase Toffoli gates, C_3Z can be written as in Eq. (57):



(57)

and likewise, C_4Z can be constructed as in Eq. (58):



(58)

This scheme—where relative phase Toffoli gates[53] are sewn together to generate a circuit for $C_{n-1}Z$ $(n > k+2)$—can be generalized. In particular, $C_{n-1}Z$ can be implemented using C_2Y, $C_2Y^\dagger$ and $C_{n-2}Z$, which in turn uses $C_{n-3}Z$. As a result of this recursion, the number of CNOTS for a $C_{n-1}Z$ circuit is

$$\#(C_{n-1}Z) = 2(n-3)\,\#(C_2Y) + \#(C_2Z). \tag{59}$$

Thus, the number of CNOTs required to implement a single query of n-qubit Grover scales as $O(n)$. At the same time, the number of necessary ancillas is $n-2$, i.e., it also scales linearly with n. As we did by using Eq. (59), this linear scaling of ancillas could be avoided by considering C_kY with $k > 2$ while increasing the number of CNOTs. Whether entangling fewer qubits by allowing for deeper circuits is worthwhile will depend on the QPU architecture under consideration. Note that the theoretically optimal number of queries $q_{\mathrm{opt}} = O(2^{n/2})$ so at q_{opt}, the number of CNOTs scales as $O(2^{n/2}n)$ where the exponential component will dominate. However, as we noted before, the experimentally allowed number of queries before decoherence takes over, might be less than q_{opt}.

DATA AVAILABILITY

The data supporting the findings of this paper is available here. https://www.dropbox.com/scl/fo/se3lsf5i20lf3j6gokt8n/h?rlkey=cfchyc0fuczs4c5hbf1pvyeq4&dl=0.

CODE AVAILABILITY

The code supporting the findings of this paper is available here. https://www.dropbox.com/scl/fo/se3lsf5i20lf3j6gokt8n/h?rlkey=cfchyc0fuczs4c5hbf1pvyeq4&dl=0.

Received: 25 January 2023; Accepted: 29 November 2023;
Published online: 20 February 2024

REFERENCES

1. Grover, L. K. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.* **79**, 325–328 (1997).
2. Bennett, C., Bernstein, E., Brassard, G. & Vazirani, U. Strengths and weaknesses of quantum computing. *SIAM J. Comput.* **26**, 1510–1523 (1997).
3. Dürr, C., Heiligman, M., Hoyer, P. & Mhalla, M. Quantum query complexity of some graph problems. *SIAM J. Comput.* **35**, 1310–1328 (2006).

4. Magniez, F., Santha, M. & Szegedy, M. Quantum algorithms for the triangle problem. *SIAM J. Comput.* **37**, 412–424 (2007).
5. Lubinski, T. et al. Application-oriented performance benchmarks for quantum computing. *IEEE Trans. Quantum Eng*. (2023).
6. Roy, T. et al. Programmable superconducting processor with native three-qubit gates. *Phys. Rev. Applied* **14**, 014072 (2020).
7. Figgatt, C. et al. Complete 3-Qubit Grover search on a programmable quantum computer. *Nat. Commun.* **8**, 1–9 (2017).
8. Zhang, K., Rao, P., Yu, K., Lim, H. & Korepin, V. Implementation of efficient quantum search algorithms on NISQ computers. *Quant. Inf. Proc.* **20**, 233 (2021).
9. Zhang, K., Yu, K. & Korepin, V. Quantum search on noisy intermediate-scale quantum devices. *Europhys. Lett.* **140**, 18002 (2022).
10. Park, G., Zhang, K., Yu, K. & Korepin, V. Quantum multi-programming for Grover's search. *Quantum Inf. Proc.* **22**, 54 (2023).
11. Vaidman, L., Goldenberg, L. & Wiesner, S. Error prevention scheme with four particles. *Phys. Rev. A* **54**, R1745–R1748 (1996).
12. Gottesman, D. *Stabilizer codes and quantum error correction*. Ph.D. thesis, California Institute of Technology (1997).
13. Kandala, A. et al. Error mitigation extends the computational reach of a noisy quantum processor. *Nature* **567**, 491–495 (2019).
14. Nachman, B., Urbanek, M., de Jong, W. A. & Bauer, C. W. Unfolding quantum computer readout noise. *npj Quantum Inf.* **6**, 84 (2020).
15. Blume-Kohout, R. et al. Robust, self-consistent, closed-form tomography of quantum logic gates on a trapped ion qubit. Preprint at http://arXiv.org/abs/1310.4492 (2013).
16. Merkel, S. T. et al. Self-consistent quantum process tomography. *Phys. Rev. A* **87**, 062119– (2013).
17. Viola, L. & Lloyd, S. Dynamical suppression of decoherence in two-state quantum systems. *Phys. Rev. A* **58**, 2733–2744 (1998).
18. Viola, L., Knill, E. & Lloyd, S. Dynamical decoupling of open quantum systems. *Phys. Rev. Lett.* **82**, 2417–2421 (1999).
19. Zanardi, P. Symmetrizing evolutions. *Phys. Lett. A* **258**, 77–82 (1999).
20. Vitali, D. & Tombesi, P. Using parity kicks for decoherence control. *Phys. Rev. A* **59**, 4178–4186 (1999).
21. Genov, G. T., Schraft, D., Vitanov, N. V. & Halfmann, T. Arbitrarily accurate pulse sequences for robust dynamical decoupling. *Phys. Rev. Lett.* **118**, 133202– (2017).
22. Khodjasteh, K. & Lidar, D. A. Fault-tolerant quantum dynamical decoupling. *Phys. Rev. Lett.* **95**, 180501 (2005).
23. Quiroz, G. & Lidar, D. A. Optimized dynamical decoupling via genetic algorithms. *Phys. Rev. A* **88**, 052306 (2013).
24. IBM Quantum. https://quantum-computing.ibm.com/ (2022).
25. Tripathi, V. et al. Suppression of crosstalk in superconducting qubits using dynamical decoupling. *Phys. Rev. Appl.* **18**, 024068 (2022).
26. Zhou, Z., Sitler, R., Oda, Y., Schultz, K. & Quiroz, G. Quantum crosstalk robust quantum control. *Phys. Rev. Lett.* **131**, 210802 (2023).
27. Boyer, M., Brassard, G., Hoyer, P. & Tapp, A. Tight bounds on quantum searching. *Fortschr. Phys.* **46**, 493–505 (1998).
28. Biham, E., Biham, O., Biron, D., Grassl, M. & Lidar, D. A. Grover's quantum search algorithm for an arbitrary initial amplitude distribution. *Phys. Rev. A* **60**, 2742–2745 (1999).
29. Blank, C., Park, D. K., Rhee, J.-K. K. & Petruccione, F. Quantum classifier with tailored quantum kernel. *npj Quantum Inf.* **6**, 1–7 (2020).
30. Rivas, A. & Huelga, S. F.*Open Quantum Systems: An Introduction*. *Springer Briefs in Physics* (Springer-Verlag, Berlin, Heidelberg, 2012).
31. Horodecki, M., Horodecki, P. & Horodecki, R. General teleportation channel, singlet fraction, and quasidistillation. *Phys. Rev. A* **60**, 1888–1898 (1999).
32. Nielsen, M. A. A simple formula for the average gate fidelity of a quantum dynamical operation. *Phys. Lett. A* **303**, 249–252 (2002).
33. Paz-Silva, G. A. & Lidar, D. A. Optimally combining dynamical decoupling and quantum error correction. *Sci. Rep.* **3**, 1530 (2013).
34. Suter, D. & Álvarez, G. A. Colloquium: protecting quantum information against environmental noise. *Rev. Mod. Phys.* **88**, 041001 (2016).
35. West, J. R., Lidar, D. A., Fong, B. H. & Gyure, M. F. High fidelity quantum gates via dynamical decoupling. *Phys. Rev. Lett.* **105**, 230503 (2010).
36. Ng, H. K., Lidar, D. A. & Preskill, J. Combining dynamical decoupling with fault-tolerant quantum computation. *Phys. Rev. A* **84**, 012305 (2011).
37. Jurcevic, P. et al. Demonstration of quantum volume 64 on a superconducting quantum computing system. *Quantum Sci. Technol.* **6**, 025020 (2021).
38. Ravi, G. S. et al. Vaqem: a variational approach to quantum error mitigation. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)* 288–303 (IEEE, 2022).
39. Pokharel, B. & Lidar, D. A. Demonstration of algorithmic quantum speedup. *Phys. Rev. Lett.* **130**, 210602 (2023).
40. Maudsley, A. A. Modified Carr–Purcell–Meiboom–Gill sequence for NMR Fourier imaging applications. *J. Magn. Reson. (1969)* **69**, 488–491 (1986).
41. Lidar, D. & Brun, T. (eds) *Quantum Error Correction* (Cambridge University Press, Cambridge, UK, 2013).
42. Ezzell, N., Pokharel, B., Tewala, L., Quiroz, G. & Lidar, D. A. Dynamical decoupling for superconducting qubits: a performance survey. *Phys. Rev. Appl.* **20**, 064027 (2023).
43. Uhrig, G. S. Keeping a quantum bit alive by optimized π-pulse sequences. *Phys. Rev. Lett.* **98**, 100504 (2007).
44. West, J. R., Fong, B. H. & Lidar, D. A. Near-optimal dynamical decoupling of a qubit. *Phys. Rev. Lett.* **104**, 130501 (2010).
45. Vuillot, C. Is error detection helpful on IBM 5Q chips?. *Quantum Inf. Comput.* **18**, 0949–0974 (2018).
46. Harper, R. & Flammia, S. T. Fault-tolerant logical gates in the IBM quantum experience. *Phys. Rev. Lett.* **122**, 080504 (2019).
47. Urbanek, M., Nachman, B. & de Jong, W. A. Error detection on quantum computers improving the accuracy of chemical calculations. *Phys. Rev. A* **102**, 022427 (2020).
48. Srinivasan, S., Pokharel, B., Quiroz, G. & Boots, B. Scalable measurement error mitigation via iterative Bayesian unfolding. Preprint at http://arXiv.org/abs/2210.12284 (2022).
49. Krantz, P. et al. A quantum engineer's guide to superconducting qubits. *Appl. Phys. Rev.* **6**, 021318 (2019).
50. Ofek, N. et al. Extending the lifetime of a quantum bit with error correction in superconducting circuits. *Nature* **536**, 441 EP (2016).
51. Ryan-Anderson, C. et al. Implementing fault-tolerant entangling gates on the five-qubit code and the color code. Preprint at http://arXiv.org/abs/2208.01863 (2022).
52. Marvian, M. & Lidar, D. A. Error suppression for Hamiltonian-based quantum computation using subsystem codes. *Phys. Rev. Lett.* **118**, 030504 (2017).
53. Maslov, D. Advantages of using relative-phase Toffoli gates with an application to multiple control Toffoli optimization. *Phys. Rev. A* **93**, 022311 (2016).
54. Arute, F. et al. Quantum supremacy using a programmable superconducting processor. *Nature* **574**, 505–510 (2019).
55. Khodjasteh, K. & Viola, L. Dynamically error-corrected gates for universal quantum computation. *Phys. Rev. Lett.* **102**, 080501 (2009).
56. Khodjasteh, K., Lidar, D. A. & Viola, L. Arbitrarily accurate dynamical control in open quantum systems. *Phys. Rev. Lett.* **104**, 090501 (2010).
57. Maciejewski, F. B., Zimborás, Z. & Oszmaniec, M. Mitigation of readout noise in near-term quantum devices by classical post-processing based on detector tomography. *Quantum* **4**, 257 (2020).
58. Pokharel, B., Anand, N., Fortman, B. & Lidar, D. A. Demonstration of fidelity improvement using dynamical decoupling with superconducting qubits. *Phys. Rev. Lett.* **121**, 220502 (2018).
59. Ronnow, T. F. et al. Defining and detecting quantum speedup. *Science* **345**, 420–424 (2014).
60. Campbell, E., Khurana, A. & Montanaro, A. Applying quantum algorithms to constraint satisfaction problems. *Quantum* **3**, 167 (2019).
61. Sanders, Y. R. et al. Compilation of fault-tolerant quantum heuristics for combinatorial optimization. *PRX Quantum* **1**, 020312 (2020).
62. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information* (Cambridge University Press, 2010).

ACKNOWLEDGEMENTS

This research was supported by the ARO MURI grant W911NF-22-S-0007. This material is also based upon work supported by the National Science Foundation, the Quantum Leap Big Idea under Grant No. OMA-1936388. We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors and do not reflect the official policy or position of IBM or the IBM Quantum team.

AUTHOR CONTRIBUTIONS

Both authors contributed equally to conceiving and designing the experiment, and to writing the manuscript. B.P. was responsible for all the data collection and analysis. D.A.L. conceived the idea and formulation of algorithmic error tomography.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41534-023-00794-6.

Correspondence and requests for materials should be addressed to Bibek Pokharel or Daniel A. Lidar.