Empirical Learning of Dynamical Decoupling on Quantum Processors

Christopher Tong,[1,2,*] Helena Zhang,[2,†] and Bibek Pokharel[2,‡]

[1] *Department of Physics, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA*
[2] *IBM Quantum, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*

(Received 20 November 2024; revised 28 March 2025; accepted 22 May 2025; published 1 August 2025)

Dynamical decoupling (DD) is a low-overhead method for quantum error suppression. Despite extensive work in DD design, finding pulse sequences that optimally decouple computational qubits is not well understood. Using IBM's superconducting-qubit-based quantum processors, we demonstrate how to empirically tailor DD strategies for an arbitrary quantum circuit and device. These learned DD strategies significantly improve error suppression relative to canonical sequences, with relative improvement increasing with problem size and circuit sophistication. We leverage this to study mirror randomized benchmarking on 100 qubits, Greenberger-Horne-Zeilinger state preparation on 50 qubits, and the Bernstein-Vazirani algorithm on 27 qubits. Our empirical learning method finds strategies, in constant time independent of circuit width and depth, provides stable performance over long time periods without retraining, and generalizes to larger circuits when trained on small subcircuit structures.

DOI: 10.1103/h7pq-s159

I. INTRODUCTION

Error suppression is vital for useful quantum computation before fault tolerance [1–4] and can continue to improve quantum computation in the fault-tolerant regime [5]. Dynamical decoupling (DD) [6–9] was one of the earliest methods to be proposed for quantum error suppression [10,11]. Numerous novel DD sequences have been proposed in the past two decades and there have been extensive theoretical studies and experimental demonstrations of DD across superconducting circuits [12–17], trapped ions [18], and solid-state electron spins [19–22]. Most DD strategies are theoretically derived assuming a specific noise model and pulse imperfection criteria. Moreover, such sequences focus on specific error suppression goals, such as the cancelation of a given term in the system-bath interaction Hamiltonian, increasing the order in the pulse spacing τ to which errors are suppressed, or reducing the effect of systematic errors in pulse implementation on a single qubit. We refer curious readers to Ref. [13] for a comprehensive survey of various DD sequences.

Even as the scale of quantum circuits used in today's experiments has grown, single-qubit sequences such as the Carr-Purcell-Meiboom-Gill (CPMG), universally robust DD (URDD) [23], and Eulerian DD (EDD) [24] sequences are often applied to multiqubit experiments [25–27]. This is despite the fact that theoretical guarantees for these sequences in canceling single-qubit errors do not extend to multiqubit quantum crosstalk [28], a central source of error in large circuits.

While multiqubit DD sequences exist, their application on quantum processors rarely succeeds [29] as multiqubit quantum gates are substantially more error prone than single-qubit ones [30]. As a compromise, staggered single-qubit DD sequences have been utilized for crosstalk suppression [28,31,32] on larger circuits. However, DD sequence staggering also introduces complications. In particular, finding sequences that cancel overlapping crosstalk terms in a quantum circuit becomes progressively nontrivial as the qubit number and circuit complexity increase. This difficulty is reflective of the challenge presented by the characterize-then-decouple strategy—where one first learns noise characteristics [30,33] of the device, the circuit structure, and the pulse imperfections and then designs the appropriate sequence. Instead, we propose a method that bypasses noise characterization by allowing for empirical feedback.

We leverage empirical feedback with classical optimization techniques, which have broadly enhanced the quantum computational toolbox to enable cutting-edge experiments [17,34–40]. Classical methods such as deep learning [41] and variational methods [42,43] have been applied to

*Contact author: lctong@mit.edu
†Contact author: helena.zhang@ibm.com
‡Contact author: bibek.pokharel@ibm.com

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

study DD, but have only provided marginal improvements. Moreover, no effective design principle to address arbitrary large-scale quantum circuits has emerged. We extend the use of classical optimization for DD by providing a generic and scalable framework to empirically learn DD strategies tailored to any given quantum device and task at hand. For our learning, we take inspiration from how Quiroz and Lidar [44] used a genetic algorithm to navigate the single-qubit DD space. Here, we introduce the genetic-algorithm based dynamical decoupling (GADD) method, which generalizes this approach by considering multiqubit circuits, while focusing on computational instead of quantum memory improvements by including empirical feedback from multiqubit circuits executed on bona fide quantum devices. Furthermore, we elaborate on how to decouple the training process from the desired quantum task, which is not necessarily classically simulable. We give a systematic description of how to train DD sequences for a given quantum circuit, even when it is large and classically intractable to simulate.

We go beyond detailing our empirical DD learning strategy by demonstrating its capabilities across three experimental settings. We demonstrate error suppression in an oracular algorithm, a state preparation circuit, and a randomized benchmarking circuit ensemble. In each case, we show that empirically trained sequences comfortably outperform canonical sequences. We also report the extension of mirror randomized benchmarking (MRB) to 100 qubits, which was enabled by learned DD sequences but not with any canonical DD strategies. Our experimental results provide compelling demonstrations of the scalability, robustness, and generalizability of empirical DD learning.

We start in Sec. II with a focus on constructing the search space of possible strategies for our empirical scheme and develop a theoretical framework for analyzing the learned DD sequences. In Sec. III, we discuss the specifics of our learning scheme. In Sec. IV, we describe our experimental results. Finally, in Sec. V, we discuss the implications of our results for DD design.

II. BACKGROUND ON DYNAMICAL DECOUPLING

Consider a simplified framework typical of the context in which DD sequences are theoretically analyzed. The unitary time evolution operator f_τ under a time-independent system-bath interaction Hamiltonian H_{SB} and bath-specific Hamiltonian H_B is given by

$$f_\tau = \exp[-i\tau(H_{\mathrm{SB}} + H_B)]. \tag{1}$$

DD sequences comprise elements of the decoupling group G, where $g_j \in G$ represents unitaries acting on the system Hilbert space $\mathcal{H}_S$. The evolution under the DD sequence $p_0, p_1, p_2, \ldots, p_L$ with time τ between each pulse is written as $U_{\mathrm{DD}}(L\tau) = p_L f_\tau p_{L-1} \cdots f_\tau p_1 f_\tau p_0$ with $\prod_j p_j = I$ [45]. Thus, we can define $q_1 = p_0$ and $q_j = q_{j-1} p_{j-1}$ for all $2 \le j \le L$, giving

$$U_{\mathrm{DD}}(L\tau) = \prod_{j=1}^{L} q_j^\dagger f_\tau q_j = q_L^\dagger f_\tau q_L \cdots q_2^\dagger f_\tau q_2\, q_1^\dagger f_\tau q_1, \tag{2}$$

where the time evolution is described by L time intervals conjugated by elements $q_j \in G$. Up to the first order in total evolution time $T = L\tau$, the DD evolution is dictated by an effective Hamiltonian H'_{SB} that satisfies

$$\begin{aligned} U_{\mathrm{DD}}(T) &= \exp\left[-i\tau \sum_{j=1}^{L} (q_j^\dagger H_{\mathrm{SB}} q_j + H_B) + O(T^2) \right] \\ &= \exp[-iT(H'_{\mathrm{SB}} + H_B) + O(T^2)]. \end{aligned} \tag{3}$$

As an example, consider the general single-qubit system-bath coupling Hamiltonian

$$H_{\mathrm{SB}} = \sum_{\alpha=x,y,z} \sigma^\alpha \otimes B^\alpha, \tag{4}$$

expressed as a sum of tensor products between Pauli operators and bath operators. If the decoupling group $G \subseteq \mathcal{P}$, the one-qubit Pauli group, then the post-DD effective system-bath coupling is

$$H'_{\mathrm{SB}} = \frac{1}{L} \sum_{\alpha=x,y,z} \left(\sum_{j=1}^{L} \mathrm{sgn}(q_j, \sigma^\alpha) \right) \sigma^\alpha \otimes B^\alpha, \tag{5}$$

where $\mathrm{sgn}(q_j, \sigma^\alpha)$ equals $+1$ if the two terms commute and -1 if the two terms anticommute. Letting $C_\alpha = \sum_{j=1}^{L} \mathrm{sgn}(q_j, \sigma^\alpha)$ gives

$$H'_{\mathrm{SB}} = \frac{1}{L} \sum_{\alpha=x,y,z} C_\alpha \sigma^\alpha \otimes B^\alpha. \tag{6}$$

For any α, $|C_\alpha|$ represents the total effective time under which the system undergoes evolution under the system-bath interaction $\sigma^\alpha \otimes B^\alpha$ to first order in τ, consistent with the summation of the switching function metric introduced in Ref. [46]. For instance, DD sequences corresponding to the $(L-1)!$ permutations of $\{g_1, g_2, g_3, g_4\} = \{I, \sigma^x, \sigma^y, \sigma^z\}$, such as the XY4 sequence, are known as first-order universal sequences as they lead to $C_\alpha = 0$ for all α. In contrast, for a CPMG sequence with only evenly spaced X pulses, $C_Y, C_Z = 0$, but $C_X \neq 0$ [13].

Generalizing to the N-qubit case, a basis for all possible correlated errors is parameterized by $\vec{\alpha} \in \{I, \sigma^x, \sigma^y, \sigma^z\}^{\otimes N}$ [47]. Then, for pulses in a decoupling group $q_j \in G \subseteq \mathcal{P}_N$,

the N-qubit Pauli group, an equivalent relationship holds, where $\sigma^{\vec{\alpha}}, q_j \in \mathcal{P}_N$ and $H_{\mathrm{SB}} = \sum_{\vec{\alpha}} \sigma^{\vec{\alpha}} \otimes B^{\vec{\alpha}}$. Then,

$$H'_{\mathrm{SB}} = \frac{1}{L} \sum_{\vec{\alpha}} C_{\vec{\alpha}} \sigma^{\vec{\alpha}} \otimes B^{\vec{\alpha}} \tag{7}$$

such that

$$C_{\vec{\alpha}} = \sum_{j=1}^{L} \prod_{k=1}^{n} \mathrm{sgn}\left(q_j, \sigma^{\alpha_k}\right). \tag{8}$$

Overall, $|C_{\vec{\alpha}}|$ quantifies the suppression offered by the DD sequence for $\sigma^{\vec{\alpha}} \otimes B^{\vec{\alpha}}$ terms to first order in evolution time T.

The number of ways to apply distinct DD sequences to each qubit in an N-qubit circuit equals $|G|^{(L-1)N}$. However, the space of DD sequences has redundancies—sequences can be sorted into equivalence classes of performance via the achieved values of $\{C_{\vec{\alpha}}\}$. For example, in the single-qubit problem, by counting the number of ways that the $L-1$ intervals where the system evolves under the effective Hamiltonian $g_j^{\dagger} H_{\mathrm{SB}} g_j$ can be partitioned among the $|G|$ possibilities for $g_{j=2,\ldots,L}$, the number of such classes is $\binom{|G|+L-2}{|G|-1}$ [48].

Although selective cancelation of system-bath interactions with DD can be much more sophisticated, as exemplified by dynamically generated decoherence-free subspaces [9,29,49], the default theoretical approach is to aim for complete cancelation such that $\sum_{\vec{\alpha}} |C_{\vec{\alpha}}| = 0$, and to use concatenation to achieve higher-order analogues [13].

In experiments, the degeneracy of equivalence classes is broken in multiple ways—due to differences in how pulses are composed and the respective imperfections. For example, on our superconducting qubits, X and Y gates are often implemented as finite-width control pulses that are error prone, while Z is implemented as a virtual state rotation that incurs no errors. This leads to distinct experimental behavior of theoretically equivalent DD sequences [12,13,50]. Moreover, while the theory above does not account for partial cancelation of errors, as we show later in the manuscript, such cancelation is useful when the number of pulses is restricted. We show that it is possible to go beyond the theoretically optimal equivalence classes of DD sequences, allowing for unconventional but high-performing DD sequences. Moreover, in a multiqubit circuit, computational performance can suffer from cross-talk and coherent errors [51], which have distinct $C_{\vec{\alpha}}$ behavior from the single-qubit case due to the product in Eq. (8). This is especially important in the generic setting where idle periods in a circuit are not aligned, making cross-talk suppression using staggered DD sequences nontrivial. Here, we design our empirical optimization search independently on k different qubit groups. Using the prior knowledge of detrimental cross-talk errors, we chose k such that any two qubits within the length scale of the dominant correlating error interaction receive independent single-qubit sequences to maximally suppress errors associated with all correlated two local $\vec{\alpha}$ on that scale. Consequently, our empirical optimization searches a space of $|G|^{(L-1)k}$ sequences. Note that, in general, it is possible to group qubits using a multidimensional hypergraph of applicable idle spaces instead of using a two-dimensional device connectivity graph originating from the connectivity of the entangling gates. Even with $k = 3$, the DD sequence space is large—for a decoupling group with $|G| = 8$, sequence length $L = 8$, and $k = 3$, there exist more than 10^{21} sequences.

Recall that theoretical attempts at evaluating DD sequences focus on averaging over the decoupling group, therefore considering only a small fraction of all possible sequences, with the goal of achieving $C_{\vec{\alpha}} = 0$ for various $\vec{\alpha}$. In this setting, the distinction between different sequences within the same decoupling group arises from robustness or higher-order cancelation properties. In contrast, our effort is focused on navigating the space of these DD sequences using an empirical feedback loop. We show that it is possible to systematically search this large space and identify well-performing DD strategies without incurring large resource overheads. In particular, by restricting our search to small k through effective, noise-informed strategies, we avoid potential exponential blowups of the search space with respect to the number of circuit qubits. By adapting to the landscape of multiqubit system-bath coherent error terms present on a quantum device, we arrive at DD sequences that decrease $|C_{\vec{\alpha}}|$ over different $\vec{\alpha}$ to increase quantum computational performance on target tasks at hand.

III. METHODS

In Sec. III A, we describe the construction of the utility function and computationally efficient training circuits as inputs for empirical learning, corresponding to Figs. 1(a), 1(b), and 1(k). The learning algorithm itself, corresponding to Figs. 1(c)–1(j), is described in Sec. III B.

A. Training circuit and utility function

To find optimal DD strategies for a given *target circuit*, our method relies on the careful selection of a *utility function* and *training circuit* such that the utility function applied on the resulting population distribution of the training circuit is an effective measure of error suppression on the target circuit. Note that the training and target circuits can but need not be identical.

In the simplest case, where the target circuit returns a single known bitstring, we can set the training circuit to the target circuit and the utility function f_U to be the success

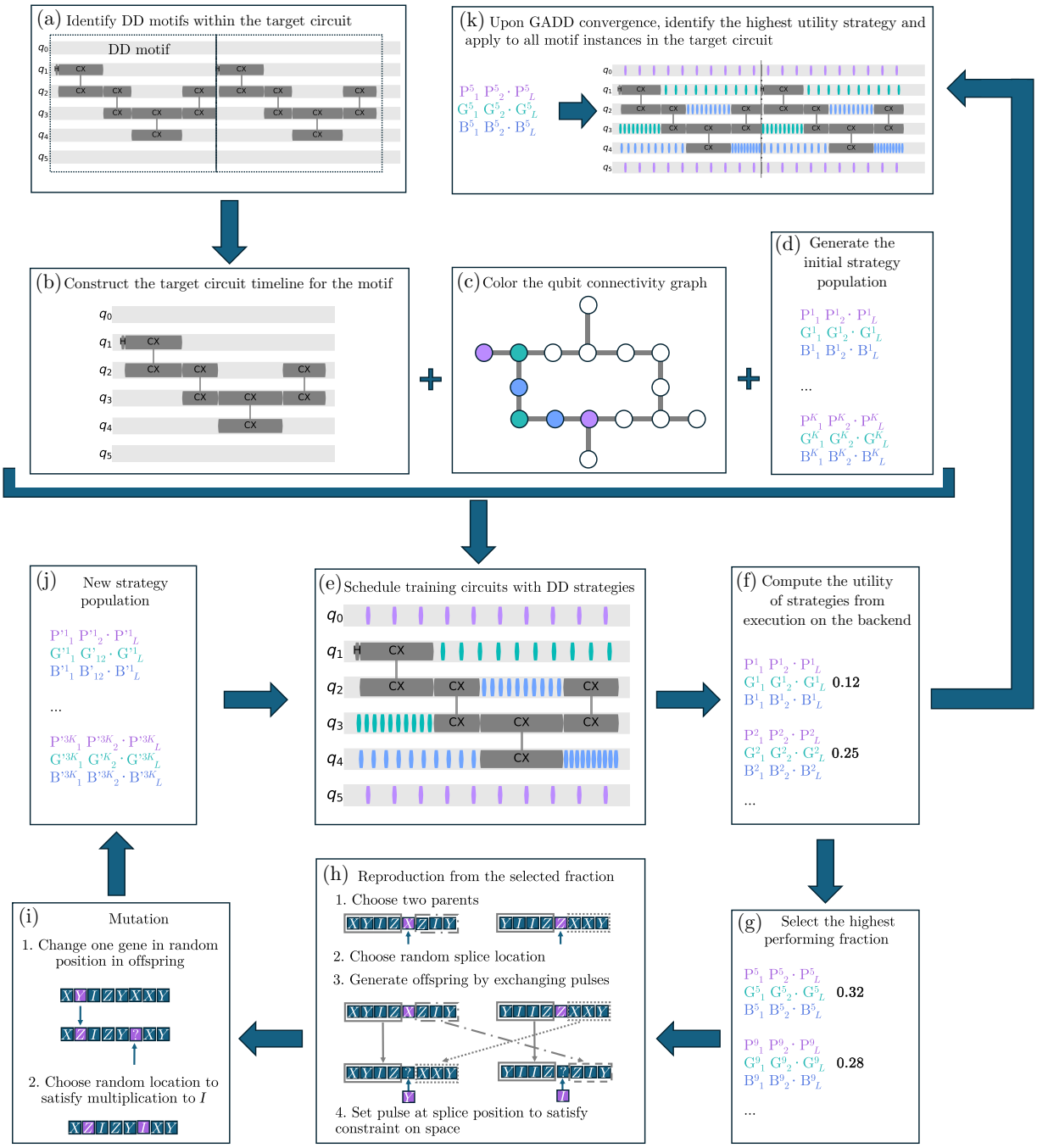


FIG. 1. Flowchart for searching for DD strategies using the genetic algorithm. After (a) identifying the motif and (b) laying out the corresponding circuit timeline, select qubits of the device are (c) tricolored into $k = \{\mathrm{P}\ (\text{purple}), \mathrm{G}\ (\text{green}), \mathrm{B}\ (\text{blue})\}$. Each individual in the initial population of size K is (d) associated with a purple green blue DD sequence that is (e) scheduled on the circuit. After utility functions are (f) computed via empirical execution on the backend, they are used to (g) select individuals for (h) reproduction. Utility functions are then recursively computed on the (j) resulting population of size $3K$ after applying (i) mutations to select the new genetic algorithm (GA) population. At any point, iteration can be terminated and the highest utility DD strategy (k) selected for experimental application.

probability p_s:

$$f_U = p_s = \frac{\text{Number of correct state measurements}}{\text{Number of total shots}}. \quad (9)$$

We demonstrate this in Sec. IV A below using the Bernstein-Vazirani problem as a first benchmark of our empirical DD search.

To generalize further, if the ideal outcome probability distribution $\mathbf{p}(c)$ of a quantum circuit is known then the

utility metric can be generalized from the success probability to the 1-norm between $\mathbf{p}(c)$ and observed $\hat{\mathbf{p}}(c)$ such that

$$f_U = 1 - \frac{1}{2}\sum_{c=0}^{2^N-1} |\mathbf{p}(c) - \hat{\mathbf{p}}(c)|. \tag{10}$$

In Sec. IV B below, we demonstrate empirical DD optimization for Greenberger-Horne-Zeilinger (GHZ) state generation using this utility function.

However, for utility-scale quantum circuits, $\mathbf{p}(c)$ is not necessarily known or hard to obtain classically. To address this challenge, we carefully construct training circuits with easily computable outcome distributions such that high-utility DD strategies on training circuits yield effective error suppression for the target circuit. We achieve this by identifying subcircuit building blocks, which we call *DD motifs*. DD motifs are subcircuit structures that capture the circuit timeline structure by accounting for idle gaps and gate operations, while ignoring the parametric details of the quantum operations. Motifs can be as simple as an idle gap on a single qubit, which the bulk of the literature in DD optimization, including Ref. [44], has focused on [13], or as complex as the entire target circuit, as discussed in the previous cases. Generally, the selection of training circuits that lie between these two extrema relies on the assumption that optimal DD strategies depend on circuit structures but not specific gate-level parameters. This is well established through prior work, such as DD studies on the motifs of neighboring idle qubits on the device graph [31], qubits adjacent to an entangling operation [28], and generic four-qubit Clifford subcircuits [42]. By explicitly introducing the abstraction of DD motifs and using it to construct sophisticated training circuits, our framework provides a pathway for DD optimization on arbitrary circuit structures.

Training circuits must have easily computable utility functions while preserving identified DD motifs. Fortunately, this leads to some flexibility in designing training circuits. For instance, DD motifs that are repeated multiple times in the target circuit can be used only once in the training circuit. Once DD motifs are identified, there exist well-known strategies for simplifying the observed probability distribution while preserving the circuit structure of the motifs. Two frequently used circuit simplification methods are circuit mirroring and Cliffordization. In circuit mirroring as proposed in Ref. [52], the output circuit is constructed by appending the inverse of each gate present in the input circuit in reverse order, maintaining the timings of all idle gaps, with symmetry-breaking gates added in between the input circuit and its inverse. Such a circuit is efficiently simulable as it behaves similarly to the identity operation under noiseless time evolution. In Cliffordization, we take the input circuit of interest, the DD motif in our case, and replace all non-Clifford gates with their Clifford counterparts, a generalization of the four-qubit DD optimization proposed in Ref. [42]. As the Clifford group acts as the stabilizer of the group of all unitary transformations on N qubits [53], the ideal outcome distribution of a Cliffordized circuit is efficiently simulable. In both cases, Eq. (9) acts as a suitable utility function of the training circuit output to learn DD strategies for the target circuit.

Keeping with the assumption that only the circuit structure must be preserved between the training and target circuits, we can identify repeating circuit substructures in the target circuit, create DD motifs on these structures to make them efficiently simulable, and then combine these DD motifs in any manner we please to create the training circuit. Put differently, the training circuit resembles the target circuit only in their common DD motifs. To our knowledge, there have been no previous attempts to compose training circuits in this manner; more importantly, it is not known if such a DD training strategy would be effective. We demonstrate the effectiveness of training DD in this manner by considering two problems of broad interest: mirror randomized benchmarking in Sec. IV C below and Grover's algorithm in Appendix C.

B. Learning algorithms

Given a training circuit and a utility function, we must also define a way to navigate the space of DD sequences, i.e., the learning algorithm. We deliberately chose discrete optimization over a continuous optimization scheme. Gate-based DD sequences can be described by many continuous parameters, such as the rotation axis and angle, duration, and pulse timings. Such parameters can be fed into continuous parameter optimization algorithms [54], such as gradient-based methods or alternative heuristic methods (e.g., Nelder-Mead), in a similar fashion to pulse-based variational strategies [43,55]. In particular, stochastic gradient descent is used in Ref. [56] to optimize a single DD sequence on a Bell state preparation circuit containing up to eight intermediate qubits. However, a continuous approach introduces a large number of parameters, and devising a principled approach to reduce these parameters for a tractable number of quantum-to-classical calls while enforcing convergence on large-scale quantum circuits is difficult. Previous empirical methods for DD optimization through deep learning, variational methods, etc. suffer from similar scalability problems, limiting them to small circuit motifs and/or low-dimensional sequence spaces, such as in Refs. [41–43,57]. Discrete optimization can allow for efficient convergence without requiring a large number of parameters or frequent context switching between quantum devices and classical processing of the results.

In order to search this large discrete space, we employ a genetic algorithm, an approach inspired by natural selection to optimize a given utility function [58,59]. We selected the genetic algorithm as the base for our approach due to the inherent parallelization in numerous independently evolving DD strategies associated with different colors that can be evaluated simultaneously. Our approach is inspired by but distinct from that of Quiroz and Lidar [44] in that we are focused on multiqubit empirical optimization using graph coloring of the device connectivity to bias our search towards cross-talk-canceling sequences. A graph-coloring approach to DD has been theoretically analyzed by Brown and Lidar [60], albeit not in the context of empirical optimization. The DD strategy space we consider is the aforementioned space of k different length-L pulse sequences drawn from a given decoupling group G on an N-qubit training circuit. Throughout this work, we refer to each of the k distinct pulse sequences as corresponding to a distinct *color*. In the prototypical genetic algorithm, an individual in a population can be identified by features analogous to genes on a chromosome. By identifying two individuals in the population and a scheme to exchange features among the two, additional individuals in the population can be generated in a process analogous to reproduction. Mutations can also be induced in individuals to increase the population diversity.

The problem of determining DD strategies for a quantum circuit admits an unambiguous representation in the context of a genetic algorithm [44]; in particular, the single-qubit pulses $\{p_j\}$ making up each length-L sequence for each of the k colors are treated as the specific genes that identify unique elements of the population of DD strategies. The structure of the decoupling group G enforces the condition that all $p_j \in G$; in this paper, we consider pulses from the group

$$G = \{I_p, I_m, X_p, X_m, Y_p, Y_m, Z_p, Z_m\}, \tag{11}$$

where I_p, X_p, Y_p, Z_p are the pulses associated with the typical Pauli matrices and I_m, X_m, Y_m, Z_m are their respective equivalents with rotation in the opposite direction on the Bloch sphere [61]. Most well-known canonical DD sequences, such as the XY4 sequence, EDD sequence, and CPMG sequence and its X_pX_m variant, are within this search space. The sequence length L is chosen such that idle periods in the target circuit are sufficiently long for the padding of L pulses. Finally, we partition the N circuit qubits into k sets by coloring a graph where nodes represent circuit qubits and edges represent entangling gates among qubits. Although k can be as large as to assign each circuit idle gap to a different color, we experimentally find that, for k sufficiently small to encompass qubits experiencing the dominant correlating interaction of ZZ crosstalk [28], effective DD sequences are learned. Thus, we use $k \leq 3$ to explore a sequence space including generalized staggered DD sequences for suppression of qubit-qubit crosstalk [62] As the IBM quantum processor's heavy hexagon architecture is planar and cannot contain any subgraph isomorphic to K_4 [63], three colors ensure that all qubits experiencing entangling gates receive independent sequences, and we further use $C = 2$ for linear qubit subgraphs. Analogous processes can be performed on other limited-connectivity architectures and easily generalize to quantum processors where alternative sources of error beyond ZZ crosstalk dominate.

We construct an initial population of K DD strategies [Fig. 1(d)], which are then inserted onto corresponding idle gaps in the given quantum circuit [Fig. 1(e)]. The resulting circuit is executed on the quantum computer for a fixed number of shots. For each strategy, we compute a utility function of the observed distribution via projective measurement of the final state given by the circuit [Fig. 1(f)]. The K highest utility strategies are selected to create the next strategy generation [Fig. 1(g)], where K pairs of individuals are selected with replacement for reproduction to produce $2K$ offspring [Fig. 1(h)], which are then subject to mutation [Fig. 1(i)]. The circuits of the combination of parents and offspring of size $3K$ are then executed. Iteration continues [Fig. 1(j)] either until a satisfactorily high-utility DD strategy is found or a set number of iterations is reached [Fig. 1(k)]. We note that, through training circuit construction and noise-informed graph coloring, DD learning with the genetic algorithm converges in $\lesssim 12$ iterations in all tested experimental settings, and the limited quantum computational training time can be amortized over many experiments (Sec. IV B below). Further discussion and simulation of GADD convergence and discussion of other DD optimization algorithms are presented in Appendix B.

IV. RESULTS AND DISCUSSION

Here we highlight the use of our learning strategy on a variety of problems. In each instance, an empirical DD search yields a positive impact on computational results. These experiments were chosen to exemplify three different choices for the utility function as well as different training settings: the success probability of the target circuit for the Bernstein-Vazirani algorithm (Sec. IV A), which is an oracular algorithm; the 1-norm of the observed probability distribution for the GHZ state preparation circuit (Sec. IV B); the success probability of smaller subcircuits from a representative random sample for MRB, a large-scale benchmarking protocol for programmable quantum computers (Sec. IV C).

A. Bernstein-Vazirani algorithm

The Bernstein-Vazirani (BV) algorithm was one of the first oracular quantum algorithms to demonstrate a

theoretical quantum speedup [64]. In its conventional form, the BV algorithm provides a linear quantum speedup over classical computation, and, recently, an algorithmic quantum speedup was demonstrated in the single-shot BV model [27] via the use of URDD for error suppression. The existence of positive results with existing DD methods, as well as the linear scaling of the BV quantum circuit with problem size and a deterministic target string, makes the BV algorithm a standard choice for benchmarking protocols on programmable quantum devices [65–67]. Here, we use the BV quantum circuit to compare the performance of GADD against a suite of well-established DD sequences commonly used in experimental settings today, including the symmetric and staggered versions of the CPMG, XY4, and EDD sequences, as well as the UR_{16} universally robust sequence. In these staggered versions, qubits of different colors experience staggered pulse timings; URDD is not staggered as this protocol selects pulse numbers based on idle duration lengths and thus has its own internal timing regulation [23,68].

The BV-n problem corresponds to the quantum task implementing an oracle encoding hidden bitstring $b \in \{0,1\}^n$, which outputs $f_b(x) = b \cdot x \pmod 2$ when queried with the input bitstring x. The goal is to identify b with the minimal number of oracle queries. The classical success probability after one iteration is $p_{s,\text{classical}} = 2/2^n$, while the ideal quantum success probability is $p_{s,\text{quantum}} = 1$ and requires $N = n + 1$ qubits to implement. While there are 2^n different oracles in the BV-n problem class, we choose to consider only $b = 1^n$ to be representative of the problem for each n as it has the largest circuit depth, following the lead of Refs. [27,65,69]. These circuits are run on the 27-qubit `ibm_peekskill` backend with the IBM Falcon architecture. We consider the problem sizes $n = 8, 10, 12, \ldots, 26$ up to the largest possible size on a 27-qubit device and repeat each circuit for 10 000 shots. The empirical learning experiment is performed with sequences of length $L = 8$ for 20 iterations and an initial population of size $K = 16$ with Eq. (9) as the utility function and 1^n as the correct state.

First, we focus our attention on the convergence of the GADD algorithm in the $n = 24$ case, which is characteristic of the learned sequences' behavior on the BV problem in the larger n regime [Fig. 2(a)]. We consider the highest success probability over strategies in the population at each iteration versus the iteration number to demonstrate the convergence of the population to one with maximum success probability 0.04 within ten iterations. This represents more than a fourfold increase in the success probability achieved with the URDD sequence and staggered CPMG sequence, while all other sequences give success probability $p_s = 0$.

The general case in the large-n regime yields similar results upon comparing GADD with benchmark sequences. We find that our empirical method strikingly outperforms all theoretically derived sequences in the test suite [Fig. 2(b)]. For comparison, in the smaller problem size regime of $n \lessapprox 16$, we find that both GADD and URDD have high success probabilities, reproducing previous demonstrations of empirically effective error suppression with URDD [13,25,27]. The relative improvement of empirically learned strategies over URDD increases with problem size, and the success probability obtained via GADD achieves a much slower decay of p_s with respect to problem size n when compared to all other DD strategies in our test suite.

We note two interesting observations in Fig. 2. First, the staggered DD sequences did not outperform their nonstaggered counterparts. As we explain in the next section, the learned sequences are able to tailor error suppression to device-specific cross-talk errors in a more precise manner than achieved by the naive staggering of sequences.

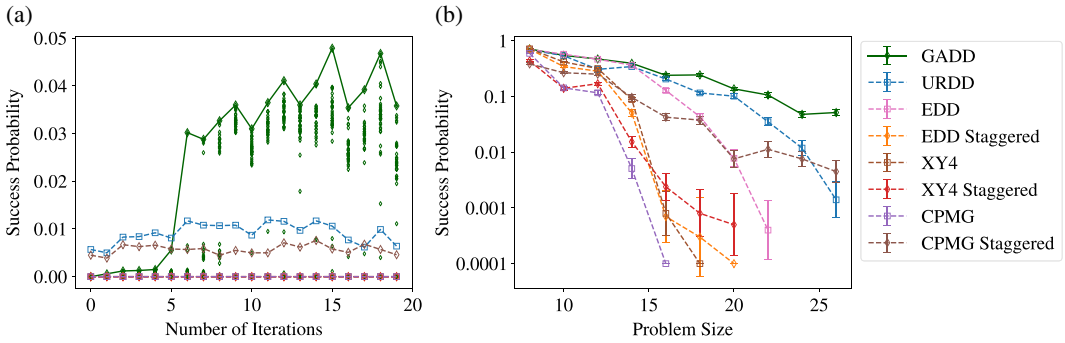


FIG. 2. Success probability for the Bernstein-Vazirani circuit on `ibm_peekskill` (a) on problem size 24 for one training session and (b) as a function of problem size for BV-8 to BV-26. GADD is compared against a suite of established DD methods. Staggered sequences are represented by open diamonds and nonstaggered sequences by open squares. Fluctuations of about 1% in the success probability are observed due to systematic device performance fluctuations between iterations. The maximum success probability of finding the 1^n bitstring is evaluated across different values of n and GADD represents a significant improvement over all other tested DD methods.

Second, robust DD sequences such as the EDD sequence did not necessarily exhibit higher computational fidelities than nonrobust sequences such as the CPMG and XY4 sequences. This is unsurprising as these sequences are only robust under certain single-qubit pulse imperfections, like flip-angle errors, which may not be the most prominent error source for the chosen device. On the other hand, URDD was constructed to be robust to a broader variety of errors, has the inherent pulse timing selection functionality, and performed quite well with respect to the other canonical methods in the suite. More importantly, empirical DD search performs well as it is able to bypass generic robustness and staggeredness requirements by optimizing suppression for the device and circuit at hand.

Although it may be possible to further improve DD performance using continuous pulse spacing optimization or hyperparameter tuning in GADD or other methods such as URDD, such improvements necessitate a computational trade-off. Additionally, improvement in computational fidelity due to DD is ultimately limited by the fact that DD sequences are only effective against non-Markovian errors [70]. Most pertinently, the demonstration that learned DD strategies can exceed the performance of canonical sequences, each of which resulted from insightful theoretical analyses, highlights the notion that DD design for physical systems can be outsourced to empirical learning algorithms.

B. Analysis of learned sequences using GHZ states

The GHZ state on N qubits, $(|0\rangle^{\otimes N} + |1\rangle^{\otimes N})/\sqrt{2^N}$, is an entangled state of great relevance in quantum computation and communication [71,72]. Preparing the GHZ state is often the first step for numerous applications such as encoding for quantum error-correcting codes [73,74], quantum cryptography [75], and Hamiltonian simulation [76,77]. As GHZ state preparation is standard among many important quantum applications, error suppression strategies for this task are of immense interest. Here, we prepare a 50-qubit GHZ state and analyze the performance of empirically learned DD strategies for the preparation of these large, highly entangled states. We further test the persistence and robustness of GADD and other canonical sequences in order to determine the extent to which empirically learned DD sequences can be reapplied for many experimental instances without incurring the overhead of additional learning. Finally, we compare the $|C_{\vec{\alpha}}|$ values achieved by learned sequences against those achieved by canonical sequences for one- and two-qubit $\vec{\alpha}$ terms to elucidate successful error suppression provided by the learned sequences.

For DD training on GHZ states, we take the 1-norm utility function [Eq. (10)], where the theoretically expected distribution is given by

$$\mathbf{p}(k) = \begin{cases} 0.5 & \text{if } k \in \{0^N, 1^N\}, \\ 0 & \text{otherwise,} \end{cases}$$

and the training circuit is taken to be identical to the 50-qubit GHZ state generation target circuit. The GADD experiment is first run for 15 iterations in the training phase. To test the persistence of these learned sequences over time, the final surviving population of the genetic algorithm is then tested for ten iterations, without any further optimization, a few hours later, and the same test is repeated with the new final population a week later. All experiments were performed on the same set of qubits on `ibm_sherbrooke`. The results are depicted in Fig. 3. In all cases, the empirically learned DD sequences perform significantly better than all tested canonical sequences and the relative DD sequence rankings remain consistent over time. We note that the difference between the empirically learned sequence and the best-performing canonical sequence, CPMG staggered, decreases over time. This is consistent with the expectation that DD sequences tailored to the device at a specific time will have decreased performance when device specifications change, shifting the landscape of single-qubit errors, multiqubit crosstalk, and any other non-Markovian errors present. However, the remaining performance improvement between empirically learned and canonical DD sequences provides evidence that we can learn to suppress many aspects of the noise environment that generalize across time, providing effectively zero application overhead to apply learned sequences subsequent to the initial learning experiment.

To further test the robustness of the learned sequences, we perform the learning experiment on one device and test the result on a different programmable device with the same underlying architecture using the same training circuit and utility function. We again perform 15 GADD iterations for empirical DD learning and apply the final population of learned sequences on `ibm_kyiv` to `ibm_sherbrooke`, on which we test the learned sequences for ten repetitions of the experiment. The results are shown in Fig. 4. Once again, the empirically discovered DD sequences outperform all the evaluated standard sequences and the relative rankings of these DD sequences remain consistent. As such, we demonstrate the transferability of empirically learned DD sequences across devices. This further supports the efficiency of DD learning, as it is not necessary to perform subsequent learning experiments for effective DD performance across devices of the same architecture.

Finally, we evaluate the underlying structure of the errors suppressed by the learned sequences by evaluating the $|C_{\vec{\alpha}}|$ terms presented in Sec. II. We use our noise-informed choice of $k = 3$ colors to suppress two-qubit cross-talk errors on connected qubits. The $|C_{\vec{\alpha}}|$ values

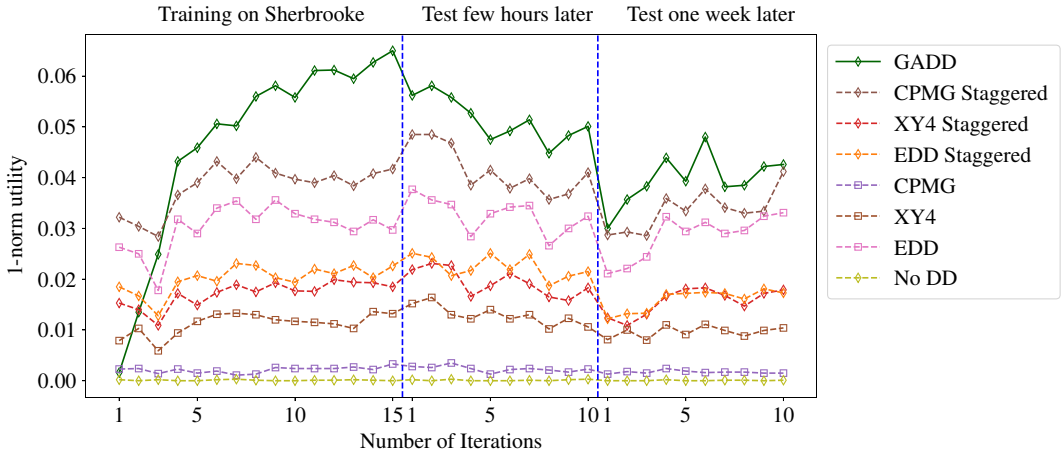


FIG. 3. Persistence of learned sequences over time. Empirical learning on the 50-qubit GHZ state preparation circuit `ibm_sherbrooke` finds DD strategies that outperform all canonical sequences in our test suite within four iterations. After a few hours, the noise environment on the quantum device does not experience significant changes, and the relative ordering of DD performance is preserved. In particular, the application of the previously learned GADD strategies leads to error suppression beyond that provided by canonical sequences without incurring any additional overhead. After one week, fluctuations in the device noise profile lead to a slight deterioration in GADD strategy performance, but the GADD sequences are still competitive and can form the starting point for the next set of training iterations.

achieved by learned DD sequences for all pairwise two-qubit $\vec{\alpha}$ terms, as well as for canonical DD sequences, are shown in Fig. 5. We chose to provide comparisons to the experimentally popular staggered CPMG sequence, currently the most frequently implemented DD sequence, and the theoretically optimal staggered EDD sequence, which is a robust sequence and has the lowest $|C_{\vec{\alpha}}|$ in the test suite. As our algorithm focuses on minimizing $|C_{\vec{\alpha}}|$ terms in the effective Hamiltonian given circuit and device restrictions and imperfections through extremizing the given utility function instead of theoretical optimization for a target $\{|C_{\vec{\alpha}}|\}$ (e.g., first-order universality, robustness, etc.), analyzing the structure of the resulting $|C_{\vec{\alpha}}|$ terms has two important implications. First, our learned sequences qualitatively differ in the partial cancelation of effective Hamiltonian terms, deprioritizing non-ZZ crosstalk. Importantly, by weighting all correlated errors according to their empirical presence on a quantum device, we arrive at *nonuniversal* single-qubit sequences that offer partial cancelation of single-qubit X and Y errors.

Second, the hierarchy of the $|C_{\vec{\alpha}}|$ terms, where we have complete cancelation of the ZZ-crosstalk and single-qubit

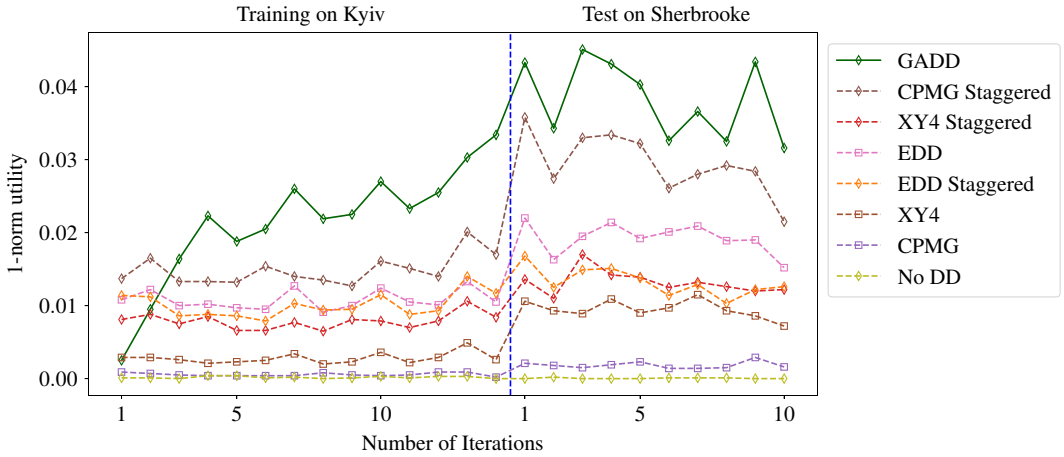


FIG. 4. Transferability of empirically learned sequences between devices. When empirically learning on the 50-qubit GHZ state preparation circuit on the `ibm_kyiv` device, we find GADD strategies that outperform all canonical sequences in our test suite within four iterations. Upon transferring the GADD strategy population on the same circuit to the `ibm_sherbrooke` device, we see similar relative performance, demonstrating the generalizability of GADD-learned strategies across superconducting devices of the same architecture.

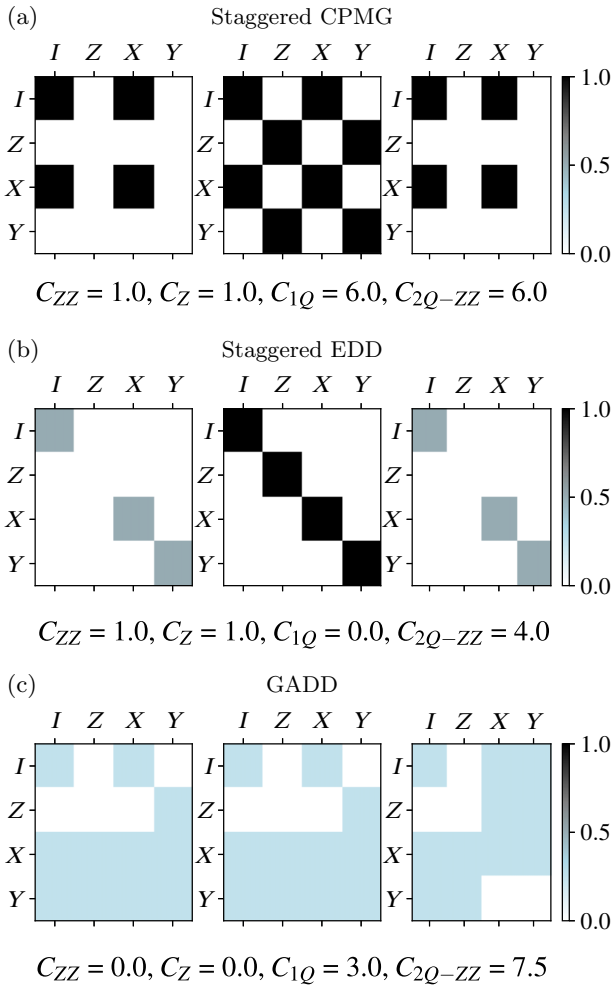


FIG. 5. Error suppression of (a) staggered CPMG, (b) staggered EDD, and (c) GADD strategies. We visualize $|C_{\vec{\alpha}}|$ for two-qubit $\vec{\alpha} = (\alpha_1, \alpha_2)$ terms by plotting α_1 on the x axis and α_2 on the y axis. Each color cell represents the normalized value of $|C_{(\alpha_1,\alpha_2)}|$ associated with k_1, k_2 (left), k_1, k_3 (middle), and k_2, k_3 (right) for our DD optimization with $k = 3$ and k_i denoting the ith color. Coefficients for the ZZ (C_{ZZ}), Z (C_Z), sum of the one-qubit coefficients C_{1Q}, and sum of the non-ZZ two-qubit terms $C_{2Q\text{-}ZZ}$ are listed. While theoretical DD sequences focus on perfect cancelation in the form of black and white squares, GADD learning arrives at fundamentally distinct optimized sequences with intermediate $|C_{\vec{\alpha}}|$ values depicted in shades of blue-gray.

Z-phase errors, followed by partial suppression of non-ZZ two-qubit errors, hints that ZZ-cross-talk and single-qubit noise terms dominate the error profile without *a priori* device calibration. Indeed, transmon qubits are known to be predominantly affected by single-qubit and ZZ-cross-talk terms [28,31]. Note that cross-talk cancelation is important but not the sole goal of error suppression—as shown in Fig. 13(b) in Appendix D, it is possible to discover optimal sequences that deprioritize certain cross-talk cancelation and still perform well. The relative values of $|C_{\vec{\alpha}}|$ help us understand the physics of device errors, but

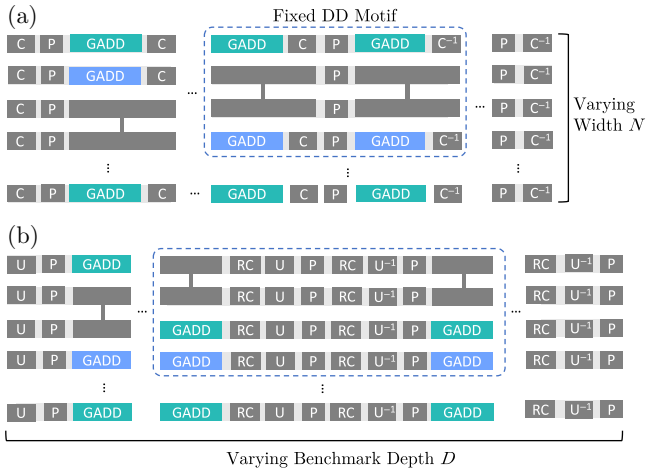


FIG. 6. Training and target circuits for (a) Clifford and (b) universal mirror randomized benchmarking [80,81]. The $D/2$ layers of one-qubit gates [C for Clifford gates and U for SU(2)] and two-qubit gate layers are mirrored across both circuits with corresponding inverses (C^{-1} and U^{-1}) for a total of D benchmark layers, and each has temporal gaps from the difference in one- and two-qubit gate times that enable the insertion of short GADD sequences. Uniformly random layers of Pauli (P) gates dress the benchmark layers, and the universal MRB single-qubit layers are further dressed with gates from randomized compiling (RC). While the overall circuits vary in the number of qubits N and D, training motifs are based on the structure of smaller fixed circuits (dotted boxes) extracted from the larger circuits and randomized to be distinct from the target circuits.

we cannot directly extract the magnitude of device noise parameters $\sigma^{\vec{\alpha}} \otimes B^{\vec{\alpha}}$ from learned sequences. Nonetheless, knowing $|C_{\vec{\alpha}}|$ can assist in the design of error suppression and calibration priorities on quantum processors. In general, even for devices that are dissimilar to those tested here, empirical optimization allows the user to bypass difficult and expensive sequence calibration and design and instead directly arrive at a sequence that works well for the task at hand.

C. Mirror randomized benchmarking

MRB is a scalable variant of the standard randomized benchmarking (RB) [78,79] protocol that characterizes the noise of quantum processors by measuring the error rates of randomized layers sampled from a defined gate set [80,81]. In Clifford MRB [80], D benchmark layers of single-qubit Clifford gates and two-qubit gates are inverted and reflected across the central time axis of the circuit. Layers of random Pauli gates dress each of the benchmark layers and a single-qubit Clifford layer and its inverse are applied at the start and end of the protocol, respectively (Fig. 6).

In the universal version of the protocol that uses gates sampled from a universal gate set such as SU(2), composite layers are constructed by randomly sampling single-qubit unitary gates from the designated gate set with a

specified density of two-qubit entangling gates, and the second half of the circuit is the inverse of the first half, randomly compiled to prevent systematic error cancelation [81]. In either protocol, mirror circuits are run on a processor and then an effective polarization S, a weighed alternative to the success probability, is calculated for each N-qubit circuit as derived in Ref. [80]:

$$S = \frac{4^N}{4^N - 1}\left[\sum_{k=0}^{N}\left(-\frac{1}{2}\right)^k h_k\right] - \frac{1}{4^N - 1}. \tag{12}$$

Here h_k is the probability that the circuit output bitstring is Hamming distance k from the target bitstring.

The error per layer (EPL) is then calculated by fitting an exponential decay curve, Ap^D, to the relation of S versus the benchmark depth D:

$$\mathrm{EPL} = \frac{2^N - 1}{2^N}(1 - p). \tag{13}$$

This corresponds to the average gate infidelity of the Pauli-dressed N-qubit benchmark layer.

While MRB is more scalable than standard RB, which is infeasible beyond five qubits [80], it still has difficulty scaling to the regime of > 50 qubits on current superconducting hardware, where other benchmarks such as layer fidelity [82] must be used instead. The difficulty is mainly due to crosstalk as the number of possible nontrivial cross-talk interactions contributing to errors increases with qubit number, while the individual gates that MRB is intended to benchmark typically have similar error rates across an entire device, leading to roughly constant EPL per qubit in the cross-talk-free regime [81,83]. As a result, there exists a threshold number of qubits for current hardware above which an effective polarization cannot be extracted even for the smallest circuit depths due to noise. We are therefore interested in examining whether it is possible to increase the scalability of MRB on real devices by reducing crosstalk using dynamical decoupling on randomized MRB circuits in the 50+-qubit regime. This would expand MRB's utility for benchmarking the infidelity of N-qubit layers on and across quantum processors.

MRB is a challenging and relevant test case for multiple reasons. First, since MRB exists in both Clifford and universal versions with similar, though not identical, circuit structures, it is a good candidate to test the validity of training on a motif that differs from the target circuit in both size and structure. If we can train on either Clifford or universal MRB circuits and find that the best GADD strategy performs well on the other problem as well, then it suggests that a technique like Cliffordization, where the training circuit is also not identical in timing to the target circuit, is likely to work on similar cases. Second, the nature of attempting to extend the benchmark to larger numbers of qubits necessitates using training circuits over smaller numbers of qubits, since the full-size target circuit will likely not have enough signal-to-noise ratio to train on. Furthermore, the condensed idle periods between layers in a typical MRB circuit are sufficiently short such that longer robust sequences cannot be padded; previous effort to improve the signal-to-noise ratio used simple CPMG sequences [83]. Lastly, the randomized nature of MRB circuits implies that the training and target circuits must be fully random and not share repeating motifs even for a constant number of qubits. Therefore, to succeed, GADD must find sequences that lower error rates on small motif circuits with relatively small idle gaps for DD insertion and also lower error rates on larger circuits with slightly different gates and timings due to the randomized nature of MRB circuits.

Taking all of the above points into account, we conduct two separate trainings, one on a single smaller motif of the Clifford MRB circuit and the other on the analogous motif of the SU(2) universal circuit, as shown in Fig. 6, then apply the learned strategy to larger circuits of both Clifford and SU(2) MRB. We perform our GADD algorithm on strategies with $L = 4$ sequences, with L limited by the relative gate density and short idle gaps of these circuits, across the `ibm_kyiv` backend. We use subsets of linear qubit arrays, which only require two different colors alternating on adjacent qubits. To train the GADD algorithm, we deliberately chose a fixed set of five random MRB circuits on ten qubits containing six circuit layers and executed each for 800 shots. We note that in addition to being a smaller circuit motif compared to the target circuit, this is significantly fewer circuits than what would be needed for a typical MRB experiment, which requires sampling sets of random circuits at different numbers of layers. For the utility function, we use the average success probability over these circuits. The echoed cross-resonance gate [84], equivalent to the controlled-NOT gate up to single-qubit prerotations, was used as the two-qubit gate of choice on the layers sampled with the edge grab sampler [52] with a two-qubit gate density of 0.25. The convergence to the asymptotic maximal utility happened quickly, within about ten iterations, and as with previous experiments, each iteration took no more than 20 s of actual quantum processor time, meaning that a training run takes fewer than 10 min on the quantum processor. The DD strategy yielding the highest success probability after training was then padded across a suite of random MRB circuits at varying benchmark depths to fit the effective polarization to an exponential and extract the EPL. The target circuits were executed without DD and with canonical XY4, EDD, and CPMG sequences, with timing staggered between the two-qubit colorings, along with the application of both GADD sequences for comparison. At each circuit width of interest, we use the exact same subarray for all experimental runs.

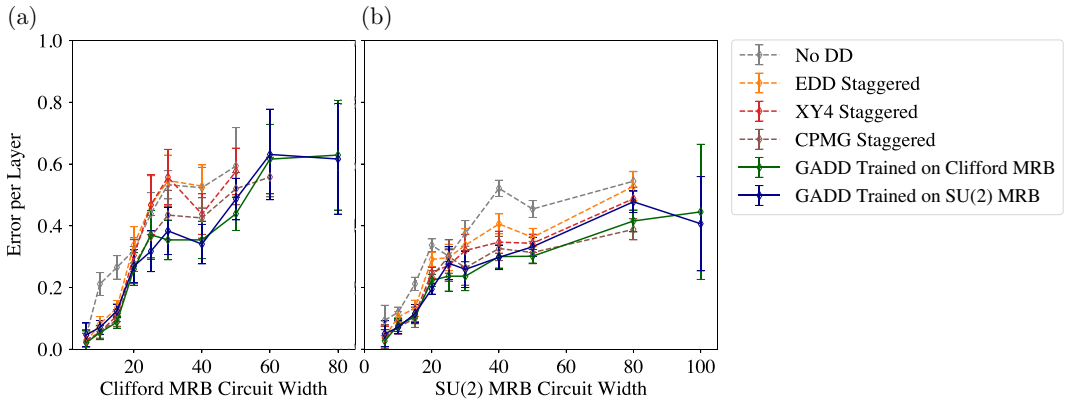


FIG. 7. Mirror randomized benchmarking results on `ibm_kyiv`. After training on a sample of small Clifford and SU(2) MRB circuits at a fixed number of circuit layers, the optimal GADD strategy is applied to target (a) Clifford MRB and (b) SU(2) MRB circuits and compared with canonical DD sequences as well as the bare circuits without DD by fitting measured polarizations at varying circuit widths and depths to obtain the EPL at each width. At larger numbers of qubits, not using DD or using suboptimal DD sequences results in polarization that cannot be fit to obtain an EPL; the data are shown for as long as an EPL can be obtained. Both GADD strategies outperform canonical DD sequences and are the only usable benchmarks at 80 qubits for the Clifford circuit and 100 qubits for the SU(2) circuit, such that MRB with GADD can be used to benchmark up to 100-qubit linear chains on this device.

The results are shown in Fig. 7. While Clifford MRB circuits beyond 50-qubit linear strings without dynamical decoupling already fail to produce any useful infidelity signal and some canonical sequences are able to reduce EPL to a certain extent, the best GADD strategies learned via training on Clifford and SU(2) circuits are both demonstrably lower in the EPL and capable of extending the range to 80 qubits [Fig. 7(a)]. The increased range also corresponds to reduced average error per layer as DD effectively reduces cross-talk contributions. This improvement extends to SU(2) MRB, where we observe that both circuits without DD and with canonical DD register lower EPL figures, but the GADD sequences are the only ones that extend to 100 qubits [Fig. 7(b)]. In Ref. [81], an experimental demonstration on a four-qubit linear array of a superconducting processor showed comparable error-per-layer rates from both Clifford and SU(2) MRB, which is similar to our results at lower numbers of qubits. It is only past about 20 qubits where the lower error rates from SU(2) MRB become evident. We hypothesize that the SU(2) MRB infidelities are broadly lower than the corresponding Clifford MRB numbers because the randomized gates themselves have decoupling properties [85], and randomly sampling gates from SU(2) provides more effective decoupling than sampling gates from a subset such as the Clifford group.

Our MRB results demonstrate the effectiveness of empirical DD learning on randomized circuits even when training on motif circuits far smaller than the desired target circuit and its potential for extending the scalability of such protocols, including, but not limited to, SU(2) universal MRB. There is no significant difference between the performance between the sequences learned from Clifford and SU(2) circuits, which suggests that, as long as the circuit structure of the motif resembles that of the final circuit, it is possible for the GADD training procedure to find a DD strategy that achieves results with lower errors and greater scalability beyond using canonical DD sequences. It shows that Cliffordization is a promising technique for using GADD on target circuits with universal gate sets different from purely Clifford training circuits. The strategy of using GADD to extend the range of effectiveness of the MRB benchmark can also be applied more broadly to other such characterization tools.

V. CONCLUSION

It is well known that error suppression on simple circuit structures, such as aligned periods of idle evolution on a block of qubits, can be optimized. However, deriving optimal sequences for general quantum circuits and devices is highly nontrivial. As a result, while a plethora of increasingly sophisticated DD sequences exist in the literature, most experimental-scale quantum computing experiments only use simple DD sequences, such as the XY4 or CPMG sequence. Here, we provide a substantial extension of previous ideas in DD optimization [28,29,42] by creating a framework to systematically and efficiently search the space of DD strategies. With the large size and relatively unknown structure of this space, matching the performance of well-studied robust sequences by empirical optimization is in itself a worthy goal. Here, we show that our learned sequences not only match but consistently outperform canonical DD strategies in error suppression. We use the switching function metric to show that learned sequences are distinctly different from canonical strategies in the assignment of weights across multi- and single-qubit errors, meaning that they cannot be easily derived from

theory. Furthermore, we demonstrate that the performance of these better-than-canonical DD sequences persists over the passage of time and the transfer between devices of similar architecture, decreasing the overhead associated with DD learning. Finally, we demonstrate the generality of our method by taking small Cliffordized motifs from large-scale MRB circuits for DD training to extend the useful range for universal MRB on a programmable quantum computer to 100 qubits.

Studying the population of optimized strategies determined by empirical DD learning can also be used to learn noise profiles and pulse imperfections of quantum devices. In particular, our method allows users to determine if and what learned sequences outperform canonical uniform multiaxis sequences. By considering patterns in the decoupling behavior, values of the $|C_{\vec{\alpha}}|$ coefficients, and inherent robustness of the learned sequences, users can determine the relative importance of suppressing specific errors on a quantum device. We hope that our work will inspire further work in using optimization techniques, perhaps beyond genetic algorithms, for navigating the large space of DD sequences to optimize error suppression. We are optimistic that such future work can be applied to discover new DD sequences of theoretical interest and acquire new insights about error suppression strategies and the physical errors affecting quantum devices.

ACKNOWLEDGMENTS

We acknowledge Luke Govia for helpful discussions and Francisco Rilloraza and Albert Zhu for developing the software to run the MRB experiments. C.T. acknowledges support from the Quantum Undergraduate Research at IBM and Princeton (QURIP) internship program. H.Z. acknowledges support from the Army Research Office under QCISS (W911NF-21-1-0002). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

DATA AVAILABILITY

The code implemented to generate the data that support the findings of this article are openly available [86]. The raw data is not publicly available but the data are available upon reasonable request from the authors.

APPENDIX A: GENETIC ALGORITHM

We begin with an initial population of DD strategies of size K. The initial population is constructed by ensuring that, for each qubit subset, each pulse appears at each site the same number of times. For example, one initial population of size $K = 16$ of $L = 8$ DD sequences with the decoupling group $G = \{I_p, I_m, X_p, X_m, Y_p, Y_m, Z_p, Z_m\}$ applied to a tricolored qubit graph consists of following set of sequences, each of which is applied to all qubit colors:

$$
\begin{aligned}
&I_p I_m X_p X_m Y_p Y_m Z_p Z_m,\\
&I_p X_p Y_p Z_p I_m X_m Y_m Z_m,\\
&I_m X_p X_m Y_p Y_m Z_p Z_m I_p,\\
&X_p Y_p Z_p I_m X_m Y_m Z_m I_p,\\
&X_p X_m Y_p Y_m Z_p Z_m I_p I_m,\\
&Y_p Z_p I_m X_m Y_m Z_m I_p X_p,\\
&X_m Y_p Y_m Z_p Z_m I_p I_m X_p,\\
&Z_p I_m X_m Y_m Z_m I_p X_p Y_p,\\
&Y_p Y_m Z_p Z_m I_p I_m X_p X_m,\\
&I_m X_m Y_m Z_m I_p X_p Y_p Z_p,\\
&Y_m Z_p Z_m I_p I_m X_p X_m Y_p,\\
&X_m Y_m Z_m I_p X_p Y_p Z_p I_m,\\
&Z_p Z_m I_p I_m X_p X_m Y_p Y_m,\\
&Y_m Z_m I_p X_p Y_p Z_p I_m X_m,\\
&Z_m I_p I_m X_p X_m Y_p Y_m Z_p,\\
&Z_m I_p X_p Y_p Z_p I_m X_m Y_m.
\end{aligned}
$$

At any given index of the pulse sequence, each element $g_n \in G$ is present at the same nonzero frequency. As a result, there is no artificial initial preference of the GA population for any individual pulse to take on any specific g_n over other elements of G.

We evaluate this method of constructing a uniform initial population by simulating the exploration of the genetic algorithm in the space of length-8 DD sequences when the utility associated with each sequence is uniformly selected at random on $[0, 1]$ to characterize how the GA explores sequences when it is agnostic of a utility function landscape. Since the DD sequence associated with any color evolves independently of other colors, this simulation is indicative of the effectiveness of the DD strategy space exploration. A simulation demonstrates that in seven GA iterations (Fig. 8), more of the DD strategy space is explored starting from the uniform population than a random starting population in expectation at any mutation probability and higher mutation probabilities lead to smaller differences between the behaviors of the two in terms of the number of unique strategies explored, as expected. This benefit is particularly important when a real optimization landscape with DD utilities is imposed, as exploring a larger fraction of the parameter space at any iteration number leads to lower probabilities of converging

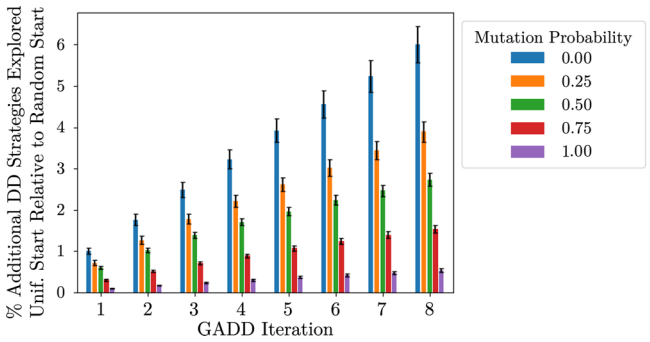


FIG. 8. The GADD algorithm with $k = 1$ was used to explore the space of DD strategies on $G = \{I_p, I_m, X_p, X_m, Y_p, Y_m, Z_p, Z_m\}$ with utility functions assigned uniformly at random to all strategies in the case of the example uniform starting population and 25 different randomly selected populations of $L = 8$ strategies. At each iteration, all pairs of DD strategies underwent reproduction to characterize the number of strategies that can theoretically be explored at that iteration number. Then, to maintain computational tractability, a random sample of 100 sequences was selected to be the new population. For each mutation probability, the average total number of DD sequences that can be explored in the random initial population case is compared relative to the average total number of DD sequences that can be explored in the uniform case. We find that at all GADD iteration numbers and mutation probabilities, starting the simulation from the example uniform starting population leads us to explore a larger amount of the DD strategy space compared to starting from a typical random population.

to local extrema that are suboptimal compared to a globally optimal DD strategy.

In each iteration of the genetic algorithm, K pairs of parent DD strategies are selected with replacement from the K individuals in the previous generation for reproduction. The probability that any individual is selected for reproduction is proportional to

$$\log 10u + 1, \tag{A1}$$

where u is the utility function of the observed count distribution associated with a DD strategy normalized to $[0, 1]$. Following the construction in Ref. [44], this has the goal of converging towards the highest-utility DD strategies, due to both the survival of high-utility DD strategies across GA iterations and the increased probability of selecting high-utility DD strategies for reproduction. After reproduction to create $2K$ offspring from the K pairs of parent strategies, the utility function is recomputed for the K elements of the previous population and the $2K$ offspring, and $K/4$ highest-utility parents and $3K/4$ highest-utility offspring are chosen for the new population to conserve the overall population size K.

The reproduction and mutation algorithms are also developed from generalizing the protocols described in Ref. [44] to the multiqubit regime. For any pair of length-L single-qubit pulse sequences, the reproduction protocol is defined as

$$c_{11}c_{12}\cdots c_{1L}, \qquad c_{21}c_{22}\cdots c_{2L} \tag{A2}$$

by selecting a random site $l \in \{1, 2, \ldots, L\}$. Then, the offspring pulse sequences can be written as

$$\begin{aligned} c'_{11}c'_{12}\cdots c'_{1L} &\equiv c_{11}c_{12}\cdots c *_{1l} c_{2(l+1)}c_{2(l+2)}\cdots c_{2L}, \\ c'_{21}c'_{22}\cdots c'_{2L} &\equiv c_{21}c_{22}\cdots c *_{2l} c_{1(l+1)}c_{1(l+2)}\cdots c_{1L}, \end{aligned} \tag{A3}$$

where $c*_{1l}$ and $c*_{2l}$ are selected to be the unique elements of G that make the resulting sequences multiply to the identity. Furthermore, we implement the single pulse where, for each offspring c', two pulse locations $1 \le l, l' \le L$ are selected. Then, c'_l is mutated to some element of G selected uniformly at random with a fixed probability, and $c'_{l'}$ is then constrained to again make the resulting sequence multiply to the identity. This protocol is extended to DD strategies on k-qubit colors by performing single-qubit reproduction and mutation on each color, followed by a randomized assignment of the two single-color offspring to the two offspring strategies.

For each new strategy generated by reproduction, the single-pulse mutation occurs with some probability that is fixed per iteration, but can vary throughout iterations of the genetic algorithm. This dynamic variation of the mutation probability is designed to accelerate the convergence of the genetic algorithm to the desired result. We find good convergence in the aforementioned results by initially setting the mutation probability to around 0.7 and dynamically increasing or decreasing the mutation probability by 0.1 if the system appears to be far away or close to equilibrium, respectively. Because of the complicated DD optimization landscape, the state of being in or out of equilibrium is difficult to quantify via iteration snapshots. We find that using proxy quantities such as the standard deviation or range of utility functions exhibited by all individuals in the population at any individual GA iteration gives reasonably good convergence results. However, such a condition is significantly dependent on the problem structure and utility function selection and there has not been sufficient rigorous testing for attribution of varying mutation rates to varying convergence.

APPENDIX B: CONVERGENCE OF GADD AND PERFORMANCE OF OTHER ALGORITHMS

We attribute the advancements in scalability of our learning strategy to three main sources. First, we presented and demonstrated in Secs. III B and IV C that our proposed training method is amenable to extending results learned on small subcircuit motifs to large-scale quantum tasks, sharply reducing the quantum computational time needed for any given problem. Second, we demonstrated in Sec. IV B that learned DD sequences persist over

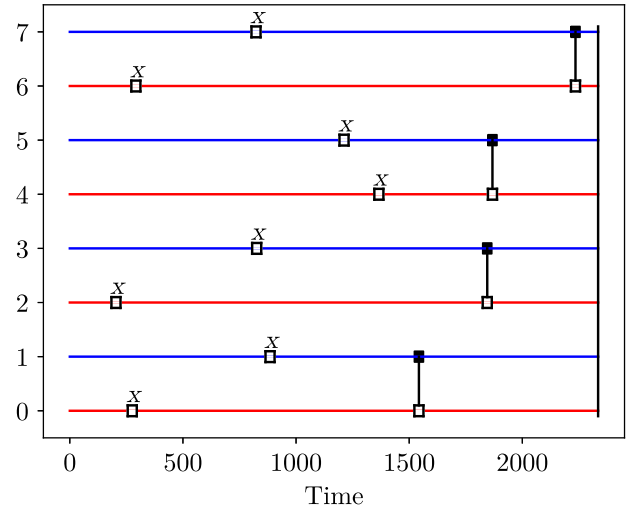


FIG. 9. Eight-qubit circuit with X gates before the pairwise entangling operation. Gate times are sampled from a uniform random distribution on $[0, 1]$: 12 random numbers are sampled and sorted in increasing order. The first eight correspond to a uniform random permutation of the X gates, while the final four correspond to a uniform random permutation of the two-qubit gates. DD sequences are placed with eight evenly spaced pulses from the circuit start to the X gate and from the X gate to the entangling operation. We evaluate the genetic algorithm performance of DD optimization with only two distinct colors (blue and red) versus with optimizing a distinct sequence for each qubit.

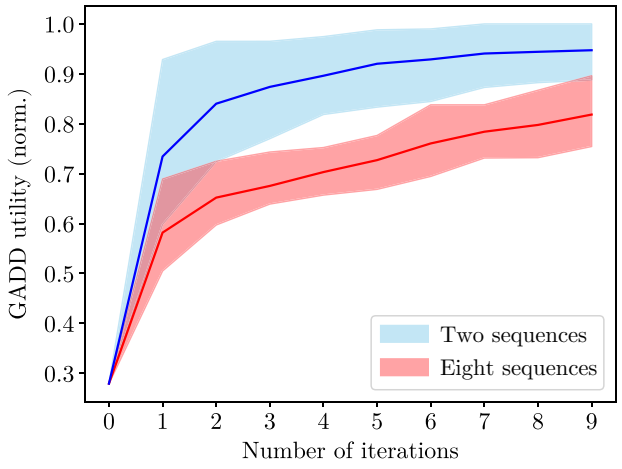


FIG. 10. Convergence of the GADD protocol over two optimized sequences (blue) and eight optimized sequences (red). The solid line represents the mean achieved maximum GADD utility for the fixed iteration, while the shaded region represents the 95% confidence interval.

weeks, allowing any training overhead to be amortized over many experimental runs. Finally, and most importantly, our strategy of coloring the device graph sharply reduces the number of degrees of freedom in the optimization problem, leading to faster and more effective convergence. In our implementation, we chose to tricolor the device graph as the leading multiqubit contribution to the error Hamiltonian on IBM superconducting qubit devices is the ZZ crosstalk between adjacent device qubits [28,31]. Furthermore, since such crosstalk is always on (i.e., not time dependent), we use a single DD sequence for all idle periods on a fixed qubit. To demonstrate the effectiveness of this approach, we simulate the GADD protocol for ten iterations for an ensemble of 100 random eight-qubit quantum circuits depicted in Fig. 9.

We simulate the GADD protocol in two settings: first, with the two coloring shown in Fig. 9, and then with one distinct sequence optimized on each of the eight qubits in parallel. We let S be the sum of $|C_{\vec{\alpha}}|$ (Sec. II) over $\vec{\alpha} = Z_{1\le i\le 8}$ and $\vec{\alpha} = Z_{2i}Z_{2i+1}$, $1 \le i \le 4$, i.e., all single-qubit phase errors and cross-talk errors, and let the GADD utility be $u = 1/S^2$. We depict the distribution of results over 100 simulations for each of the random circuits in both cases in Fig. 10.

We note that, although the eight-sequence case can lead to theoretically higher utilities due to having more degrees of freedom and thus more tunable knobs to reach an absolute minimum in utility, we observe near-ideal utilities by simply optimizing over two sequences. The trade-off in decreasing the sequence number for faster convergence certainly holds in this setting and allows for scaling to 50–100 qubits, as we demonstrated in Secs. IV B and IV C. This is also representative of the broader tunability of our method—the choice of adjacent qubit coloring and uniformity over time for a fixed qubit is noise inspired by the ZZ crosstalk on current IBM hardware. As time progresses, different coloring methods can be adapted to new leading sources of error, which can have markedly different correlations, and suppress them efficiently as long as they exist on the timescale of DD-targeted errors.

APPENDIX C: CLIFFORDIZED CIRCUITS AND GROVER'S ALGORITHM

We implement Cliffordization to preserve DD motifs present in a target circuit while reducing it to a training circuit that is efficiently classically simulable. Generically, for quantum gate U imparting phase $0 \le \phi \le \pi/2$, ϕ is set to $\pi/2$ with probability $\phi/(\pi/2)$ and 0 with the remaining probability. Analogously, this conversion is performed for $\phi \in (\pi/2, \pi), (\pi, 3\pi/2), (3\pi/2, 2\pi)$ with their respective bounds. If U also imparts a bit flip, the bit flip is converted into a phase rotation by the Hadamard gate, the previous protocol is applied to the phase rotation, and the resulting Clifford phase rotation is converted back into a bit flip with another Hadamard gate. For example, the $T^\dagger$ gate imparting a $-\pi/4$ phase on a quantum state in the computational basis is Cliffordized by replacing it with a

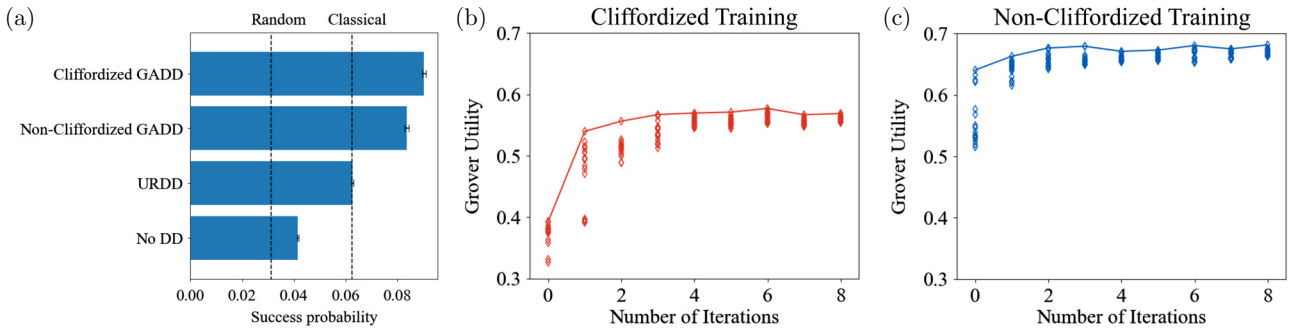


FIG. 11. (a) Five-qubit Grover algorithm's success probabilities averaged over all 2^5 oracles and the utility versus iteration for GADD training on (b) Cliffordized and (c) non-Cliffordized, or the original, Grover circuits on `ibmq_mumbai`. The GADD strategy was trained with the utility function relating to the difference from the ideal distribution as classically simulated. In both the Cliffordized and non-Cliffordized training cases, we see the convergence of the GA within about five iterations. We note that, for the non-Cliffordized circuit, our initial strategy population happened to contain high-utility individuals. The average success probability in (a) of GADD strategies from Cliffordized and non-Cliffordized training is higher than the URDD strategy and no-DD strategy. The random threshold is established from a simple random uniform guess over all 2^5 bitstrings, while the best classical strategy allows for a single classical query before guessing.

gate with $\phi = 3\pi/2$ or the identity with $\phi = 0$ with equal probability.

Grover's search algorithm provides an ideal platform to evaluate the concept of Cliffordized training. Grover's search algorithm seeks to find a hidden length-N bitstring b in an unsorted list with 2^N elements. The oracle in Grover's algorithm returns $f_b(x) = \delta_{x,b}$ when queried with the input bitstring x. Classically, searching such an unsorted list requires $O(N)$ queries. By encoding the problem in an N-qubit circuit, a quadratic improvement in the number of queries is provably achieved relative to the best-possible classical algorithm [87]. However, even for relatively small N, the circuit depths associated with even the most efficient implementations of Grover's algorithm do not exhibit better-than-classical results without DD due to errors [26].

Grover's search algorithm provides an ideal platform to evaluate the concept of Cliffordized training for two reasons. First, there are 2^N Grover search problems on N qubits that can be easily accessed by changing specific parameters in non-Clifford gates present in the Grover oracle, but without changing the general circuit structure. Second, the depth of the Grover circuit imposes the requirement that, although the $N = 5$ case poses challenges in terms of error on the quantum device, it is sufficiently small for individual non-Clifford Grover circuits to be tractably simulated classically. As a result, the effects of training with a Cliffordized circuit can be effectively compared with the effects of training with the circuit before Cliffordization.

We begin by considering two different training circuits for GADD. First, we consider the Cliffordized version of the Grover circuit encoding target bitstring 11111, as well as its original non-Cliffordized form as a control to evaluate the effect of Cliffordization. For each of these circuits, we run the GADD algorithm for DD strategies with the same hyperparameters as in the BV problem, except with utility function $u(t)$ that considers the total absolute difference between the simulated and experimentally estimated distributions of output counts in the computational basis with $N = 5$ qubits as described in Eq. (10) over 4000 shots. We note that this utility function has no knowledge of what the target string is in the Cliffordized training case. The resulting learned DD sequence is then padded to all 32 distinct circuits for $N = 5$ Grover search problems and the average search success probability over the 32 circuits is calculated. These circuits are run on the 27-qubit `ibmq_mumbai` backend.

In both the Clifford and non-Clifford training cases, we observe convergence in the population generated by GADD after about four GA iterations (Fig. 11). After implementing these learned high-utility DD strategies on all $N = 5$ Grover circuits, we demonstrate that significantly better-than-classical success probabilities are achieved under the implementation of GADD under both training cases, as depicted in the right panel of Fig. 11. In contrast, this is not achieved in the case where URDD, the strategy that generically gave the next highest performance in Sec. IV A, is implemented. Notably, the Grover success probability is comparable in the Cliffordized and non-Cliffordized training cases, which provides empirical evidence for the effectiveness of GADD training on a result-agnostic utility function. As the Cliffordized circuit provides a more general representation of the $N = 5$ qubit Grover problem over all Grover oracles, while the non-Clifford circuit preserves the nongeneralizable information of 11111 being the target string, the comparable resulting Grover success probabilities imply that encoding such output-dependent information in the GADD utility function is unimportant.

Thus, the Cliffordization method for generating training circuits is quite promising for learning successful DD strategies in the general setting where the resulting quantum state from the problem at hand is *a priori* unknown.

APPENDIX D: ANALYSIS OF LEARNED SEQUENCES

The top-three sequences when training over the 50-qubit GHZ circuits for $k = 3$ colors k_1, k_2, k_3 on `ibm_kyiv` were

$$k_1 = Z_m Y_m X_p Y_p Z_m I_m Y_p Z_m, \quad k_2 = X_p X_p Y_m I_m I_p Z_p Y_p I_m, \quad k_3 = X_m X_p X_m X_p X_m Z_m Z_m Y_m,$$
$$k_1 = I_p Y_m X_p X_p Z_m I_m Y_p Z_m, \quad k_2 = X_p X_p Y_m I_m I_m Z_p Y_p I_m, \quad k_3 = X_m X_p X_m X_p X_m Z_m Z_m Y_m,$$
$$k_1 = I_p Y_m X_p X_p Z_p I_p Y_p Z_m, \quad k_2 = X_p X_p Y_m I_m I_m Z_p Y_p I_m, \quad k_3 = X_m X_p X_m X_p X_p Z_m Z_m Y_m.$$

Likewise, the top-three sequences for `ibm_sherbrooke` were

$$k_1 = I_p Y_m I_m X_p Z_m X_m Y_p Z_m, \quad k_2 = I_p X_p Y_m X_m I_p Z_p I_p I_m, \quad k_3 = X_p Y_p X_m X_p Y_p Z_m Z_m I_p,$$
$$k_1 = I_p Y_m X_p X_p Z_m Y_p Y_p Z_p, \quad k_2 = X_p X_p Y_m Y_p I_m Z_m Y_m I_m, \quad k_3 = X_m X_p X_m X_p X_p Z_m Z_p Y_m,$$
$$k_1 = I_p Y_p X_p X_p Z_p Y_p Y_m Z_p, \quad k_2 = X_p X_p Y_p Y_p I_p Z_m Y_m I_m, \quad k_3 = Y_p X_p X_p Y_p X_p Z_m Z_m Y_m.$$

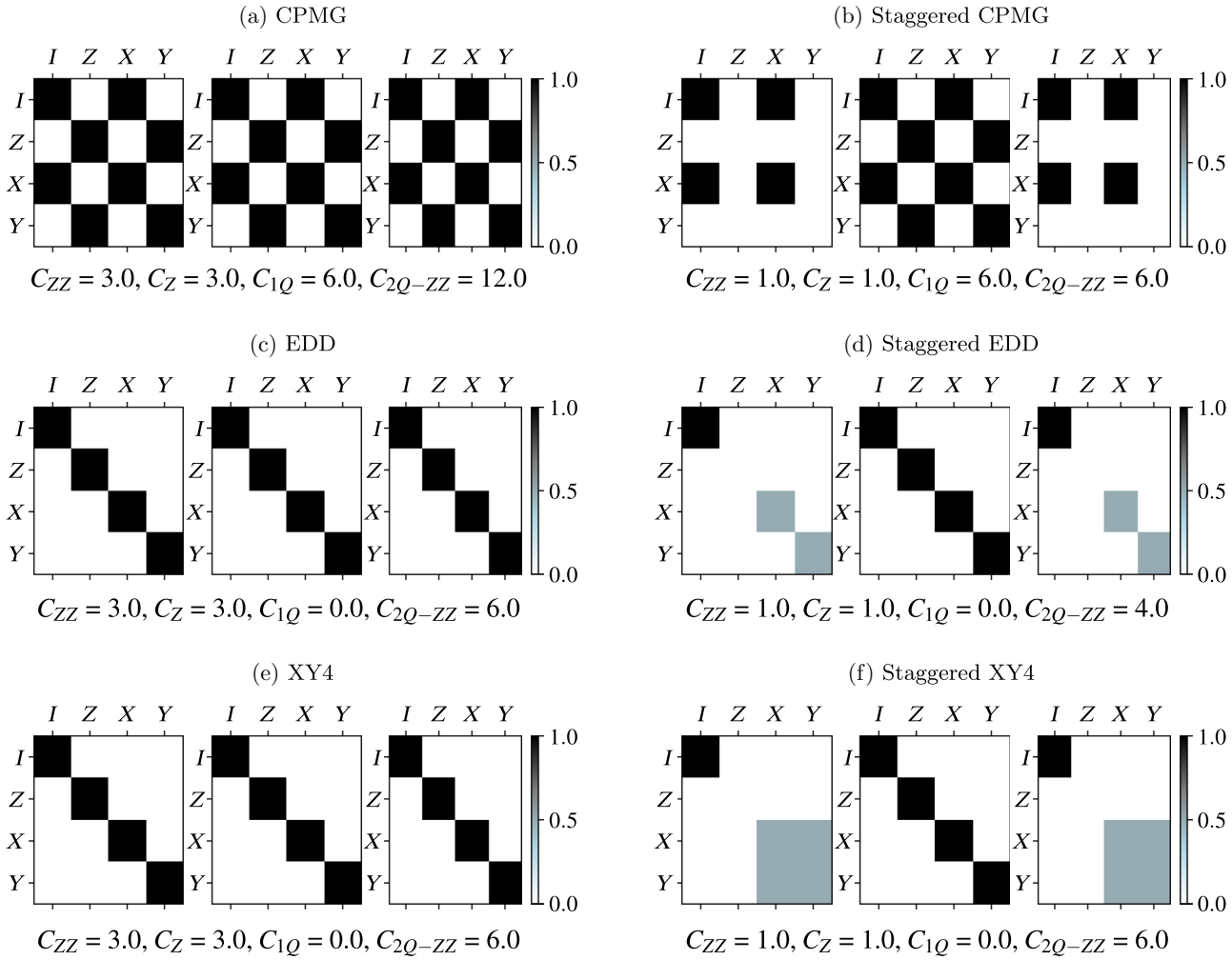


FIG. 12. Normalized $|C_{\vec{\alpha}}|$ for canonical DD sequences (a),(c),(e) CPMG, EDD, XY4 and (b),(d),(f) their staggered versions.

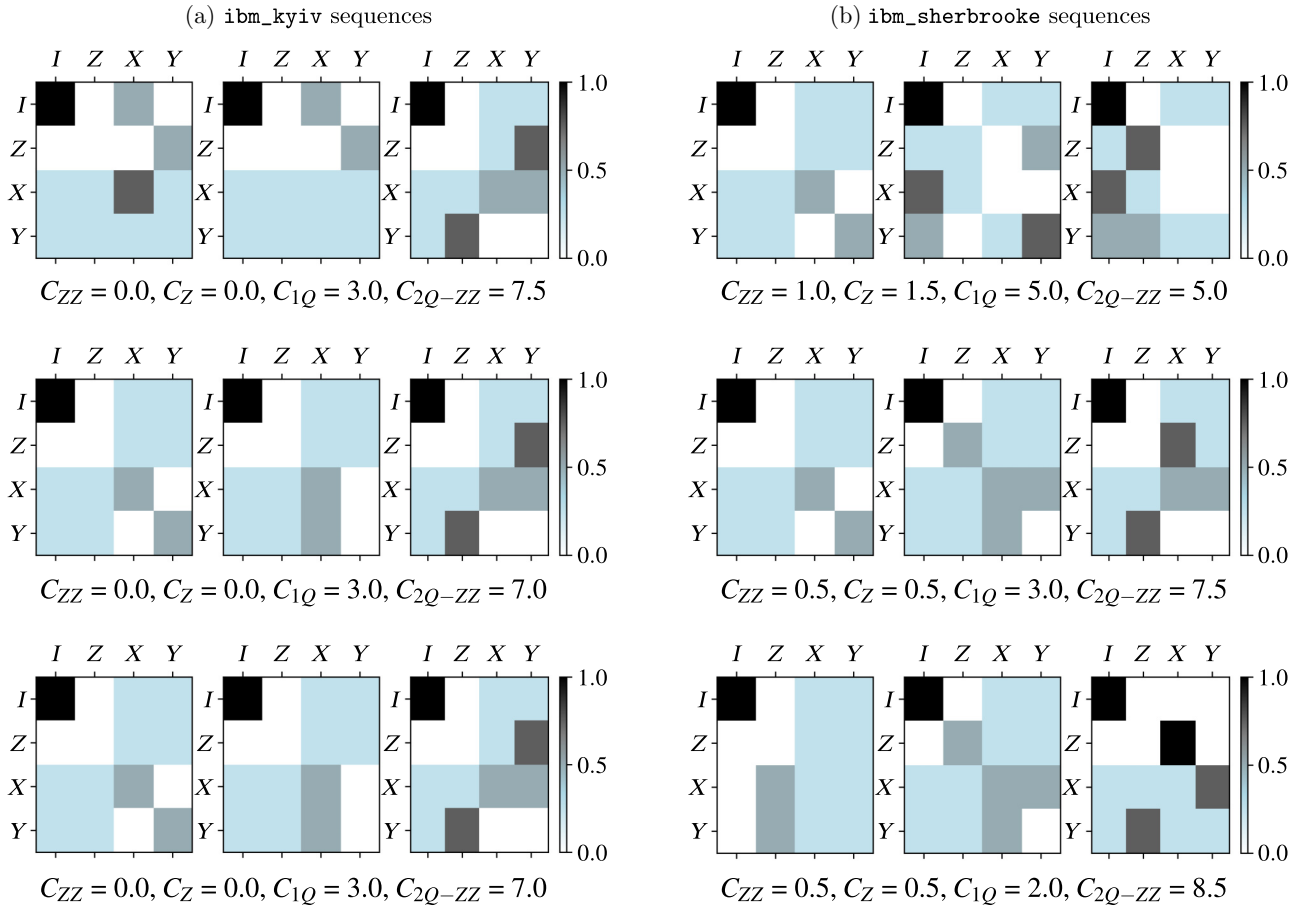


FIG. 13. Normalized $|C_{\vec{\alpha}}|$ for learned DD sequences when optimizing for the 50-qubit GHZ states. The top-three best-performing sequences from (a) `ibm_kyiv` and (b) `ibm_sherbrooke` are shown with better sequences nearer the top.

Figure 12 shows the coefficients for canonical DD sequences. In contrast, Fig. 13 shows that most of the sequences prioritize canceling the *ZZ* crosstalk. However, the learned sequences on `ibm_sherbrooke` do not completely cancel the *ZZ* crosstalk between all the qubits—crosstalk between qubits with colors 1 and 2 are always canceled, but the rest are only partially suppressed. Instead, the optimization places more emphasis on canceling the single-qubit and non-*ZZ* two-qubit error terms.

APPENDIX E: HARDWARE AND SOFTWARE SPECIFICATIONS

All experimental results in this work were obtained through Qiskit [88]. The code for MRB circuits and

TABLE I. Device specifications for `ibm_peekskill`, `ibmq_mumbai`, and `ibm_kyiv` at the time of their respective experiment executions are shown. Grover v1 and v2 refer to the Cliffordized and non-Cliffordized runs, respectively. 1QG, 2QG, and RO denote one-qubit gate, two-qubit gate, and readout, respectively.

	Peekskill (BV)			Mumbai (Grover v1)			Mumbai (Grover v2)			Kyiv (MRB)		
	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
T_1 (μs)	139.66	321.63	516.35	57.32	113.65	174.98	21.17	141.42	343.44	37.4	277.7	465.4
T_2 (μs)	23.66	299.26	536.11	27.06	167.27	343.3	27.06	167.27	343.3	4.0	120.3	506.0
1QG error (%)	0.01	0.02	0.05	0.01	0.03	0.19	0.01	0.03	0.1	0.01	0.03	0.49
2QG error (%)	0.39	0.8	1.37	0.49	0.91	2.1	0.49	0.91	1.76	0.37	1.1	100.0
1QG duration (μs)	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05
2QG duration (μs)	0.43	0.6	0.62	0.25	0.45	0.74	0.25	0.45	0.74	0.56	0.56	0.56
RO error (%)	0.37	1.97	9.59	1.0	2.84	8.02	1.0	2.84	8.02	0.14	0.88	28.82
RO duration (μs)	0.86	0.86	0.86	3.58	3.58	3.58	3.58	3.58	3.58	1.24	1.24	1.24

TABLE II. Device specifications for `ibm_sherbrooke` and `ibm_kyiv` at the time of experiment execution for GHZ state preparation experiments are shown. 1QG, 2QG, and RO denote one-qubit gate, two-qubit gate, and readout, respectively.

	Sherbrooke (GHZ, 2024-06-13)			Sherbrooke (GHZ, 2023-06-19)			Kyiv (GHZ, 2024-06-13)		
	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
T_1 (μs)	23.19	265.79	475.14	50.15	257.87	484.34	33.34	277.95	465.45
T_2 (μs)	15.64	172.51	394.96	16.93	174.29	516.61	15.52	153.08	542.36
1QG error (%)	0.01	0.04	0.62	0.01	0.04	0.62	0.01	0.04	0.67
2QG error (%)	0.33	5.13	100.00	0.31	5.04	100.00	0.34	1.3	6.3
1QG duration (μs)	0.06	0.06	0.06	0.06	0.06	0.06	0.05	0.05	0.05
2QG duration (μs)	0.34	0.54	0.88	0.34	0.54	0.88	0.56	0.56	0.56
RO error (%)	0.27	2.79	34.77	0.35	2.68	38.16	0.07	1.64	19.97
RO duration (μs)	1.24	1.24	1.24	1.24	1.24	1.24	1.24	1.24	1.24

analysis was written in Qiskit Experiments [89]. To quickly iterate in the genetic algorithm training process, Qiskit Runtime [90] was used to interleave quantum training circuits and the classical learning algorithm to evaluate the strategies.

BV, Grover, and MRB experiments were respectively run on the following IBM superconducting quantum processors: `ibm_peekskill` and `ibmq_mumbai`, which are 27-qubit Falcon r5.10 and r8 processors, respectively; `ibm_kyiv`, which is a 127-qubit Eagle r3 processor. The machine specifications for all three devices for the data reported in this work are shown in Table I. GHZ experiments were run on `ibm_kyiv` and `ibm_sherbrooke`, another 127-qubit Eagle r3 processor, with device specifications shown in Table II. Note that in certain cases the maximum two-qubit errors are at 100% due to certain qubits being out of commission; in such cases, we transpiled our circuit to avoid those qubits.

[1] P. W. Shor, Scheme for reducing decoherence in quantum computer memory, Phys. Rev. A **52**, R2493 (1995).

[2] A. M. Steane, Error correcting codes in quantum theory, Phys. Rev. Lett. **77**, 793 (1996).

[3] P. Shor, in *Proceedings of 37th Conference on Foundations of Computer Science* (IEEE, Burlington, VT, USA, 1996), pp. 56–65.

[4] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[5] H. K. Ng, D. A. Lidar, and J. Preskill, Combining dynamical decoupling with fault-tolerant quantum computation, Phys. Rev. A **84**, 012305 (2011).

[6] L. Viola, E. Knill, and S. Lloyd, Dynamical decoupling of open quantum systems, Phys. Rev. Lett. **82**, 2417 (1999).

[7] L. Viola and S. Lloyd, Dynamical suppression of decoherence in two-state quantum systems, Phys. Rev. A **58**, 2733 (1998).

[8] D. Vitali and P. Tombesi, Using parity kicks for decoherence control, Phys. Rev. A **59**, 4178 (1999).

[9] P. Zanardi, Symmetrizing evolutions, Phys. Lett. A **258**, 77 (1999).

[10] H. Y. Carr and E. M. Purcell, Effects of diffusion on free precession in nuclear magnetic resonance experiments, Phys. Rev. **94**, 630 (1954).

[11] S. Meiboom and D. Gill, Modified spin-echo method for measuring nuclear relaxation times, Rev. Sci. Instrum. **29**, 688 (1958).

[12] B. Pokharel, N. Anand, B. Fortman, and D. A. Lidar, Demonstration of fidelity improvement using dynamical decoupling with superconducting qubits, Phys. Rev. Lett. **121**, 220502 (2018).

[13] N. Ezzell, B. Pokharel, L. Tewala, G. Quiroz, and D. A. Lidar, Dynamical decoupling for superconducting qubits: A performance survey, Phys. Rev. Appl. **20**, 064027 (2023).

[14] P. Jurcevic *et al.*, Demonstration of quantum volume 64 on a superconducting quantum computing system, Quantum Sci. Technol. **6**, 025020 (2021).

[15] G. Q. AI, Suppressing quantum errors by scaling a surface code logical qubit, Nature **614**, 676 (2023).

[16] Y. Kim, C. J. Wood, T. J. Yoder, S. T. Merkel, J. M. Gambetta, K. Temme, and A. Kandala, Scalable error mitigation for noisy quantum circuits produces competitive expectation values, Nat. Phys. **19**, 752 (2023).

[17] Y. Kim, A. Eddins, S. Anand, K. Wei, E. Van Den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme, and A. Kandala, Evidence for the utility of quantum computing before fault tolerance, Nature **618**, 500 (2023).

[18] M. J. Biercuk, H. Uys, A. P. VanDevender, N. Shiga, W. M. Itano, and J. J. Bollinger, Experimental Uhrig dynamical decoupling using trapped ions, Phys. Rev. A **79**, 062324 (2009).

[19] J. Du, X. Rong, N. Zhao, Y. Wang, J. Yang, and R. B. Liu, Preserving electron spin coherence in solids by optimal dynamical decoupling, Nature **461**, 1265 (2009).

[20] G. de Lange, Z. H. Wang, D Riste, V. V. Dobrovitski, and R Hanson, Universal dynamical decoupling of a single solid-state spin from a spin bath, Science **330**, 60 (2010).

[21] Z. Wang, G. de Lange, D. Ristè, R. Hanson, and V. V. Dobrovitski, Comparison of dynamical decoupling protocols for a nitrogen-vacancy center in diamond, Phys. Rev. B **85**, 155204 (2012).

[22] D. Farfurnik, A. Jarmola, L. M. Pham, Z. H. Wang, V. V. Dobrovitski, R. L. Walsworth, D. Budker, and N. Bar-Gill, in *Quantum Optics* (SPIE, Brussels, Belgium, 2016), Vol. 9900, pp. 111–120.

[23] G. T. Genov, D. Schraft, N. V. Vitanov, and T. Halfmann, Arbitrarily accurate pulse sequences for robust dynamical decoupling, Phys. Rev. Lett. **118**, 133202 (2017).

[24] L. Viola and E. Knill, Robust dynamical decoupling of quantum systems with bounded controls, Phys. Rev. Lett. **90**, 037901 (2009).
[25] P. Singkanipa, V. Kasatkin, Z. Zhou, G. Quiroz, and D. A. Lidar, Demonstration of algorithmic quantum speedup for an Abelian hidden subgroup problem, Phys. Rev. X (2025)
[26] B. Pokharel and D. A. Lidar, Better-than-classical Grover search via quantum error detection and suppression, npj Quantum Inf. **10**, 23 (2024).
[27] B. Pokharel and D. A. Lidar, Demonstration of algorithmic quantum speedup, Phys. Rev. Lett. **130**, 210602 (2023).
[28] V. Tripathi, H. Chen, M. Khezri, K. Yip, E. Levenson-Falk, and D. A. Lidar, Suppression of crosstalk in superconducting qubits using dynamical decoupling, Phys. Rev. Appl. **18**, 024068 (2022).
[29] G. Quiroz, B. Pokharel, J. Boen, L. Tewala, V. Tripathi, D. Williams, L.-A. Wu, P. Titum, K. Schultz, and D. Lidar, Dynamically generated decoherence-free subspaces and subsystems on superconducting qubits, Rep. Prog. Phys. **87**, 097601 (2024).
[30] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, A quantum engineer's guide to superconducting qubits, Appl. Phys. Rev. **6**, 021318 (2019).
[31] Z. Zhou, R. Sitler, Y. Oda, K. Schultz, and G. Quiroz, Quantum crosstalk robust quantum control, Phys. Rev. Lett. **131**, 210802 (2023).
[32] L. Shirizly, G. Misguich, and H. Landa, Dissipative dynamics of graph-state stabilizers with superconducting qubits, Phys. Rev. Lett. **132**, 010601 (2024).
[33] H. Zhang, B. Pokharel, E. M. Levenson-Falk, and D. Lidar, Predicting non-Markovian superconducting-qubit dynamics from tomographic reconstruction, Phys. Rev. Appl. **17**, 054018 (2022).
[34] A. D. Córcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, Error mitigation extends the computational reach of a noisy quantum processor, Nature **567**, 491 (2019).
[35] E. Van Den Berg, Z. K. Minev, A. Kandala, and K. Temme, Probabilistic error cancellation with sparse Pauli–Lindblad models on noisy quantum processors, Nat. Phys. **19**, 1116 (2023).
[36] K. Temme, S. Bravyi, and J. M. Gambetta, Error mitigation for short-depth quantum circuits, Phys. Rev. Lett. **119**, 180509 (2017).
[37] P. D. Nation, H. Kang, N. Sundaresan, and J. M. Gambetta, Scalable mitigation of measurement errors on quantum computers, PRX Quantum **2**, 040326 (2021).
[38] E. Van Den Berg, Z. K. Minev, and K. Temme, Model-free readout-error mitigation for quantum expectation values, Phys. Rev. A **105**, 032620 (2022).
[39] B. Pokharel, S. Srinivasan, G. Quiroz, and B. Boots, Scalable measurement error mitigation via iterative Bayesian unfolding, Phys. Rev. Res. **6**, 013187 (2024).
[40] B. Yang, R. Raymond, and S. Uno, Efficient quantum readout-error mitigation for sparse measurement outcomes of near-term quantum devices, Phys. Rev. A **106**, 012423 (2022).
[41] A. Zlokapa and A. Gheorghiu, A deep learning model for noise prediction on near-term quantum devices, arXiv:2005.10811.
[42] P. Das, S. Tannu, S. Dangwal, and M. Qureshi, in *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture* (Association for Computing Machinery (ACM), Athens, Greece, 2021), pp. 950–962.
[43] G. K. Ravi, K. N. Smith, P. Gokhale, A. Mari, N. Earnest, A. Javadi-Abhari, and F. T. Chong, in *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)* (IEEE Computer Society, Republic of Korea, 2022), pp. 288–303.
[44] G. Quiroz and D. A. Lidar, Optimized dynamical decoupling via genetic algorithms, Phys. Rev. A **88**, 052306 (2013).
[45] D. A. Lidar, in *Quantum Information and Computation for Chemistry* (John Wiley & Sons, Ltd, Hoboken, New Jersey, 2014), Chap. 11, pp. 295–354.
[46] G. S. Uhrig and D. A. Lidar, Rigorous bounds for optimal dynamical decoupling, Phys. Rev. A **82**, 012301 (2010).
[47] For convenience, we include the case where $\vec{\alpha} = \{I, I, \ldots, I\}$, although it is irrelevant whether it is included or not as such a term only acts on the system via a time-varying global phase.
[48] This corresponds to the combinatorial problem for which we count the number of ways $L-1$ indistinguishable objects can be arranged in $|G|$ distinguishable bins. The $L-1$ objects correspond to the g_j operators, which are indistinguishable because the first order $|C_{\vec{\alpha}}|$ is invariant to any permutation of the $j = 2, \ldots, L$, while the $|G|$ bins correspond to elements of the decoupling group, which have distinct commutation relations with H_{SB} and are thus distinguishable.
[49] L. Wu and D. A. Lidar, Creating decoherence-free subspaces using strong and fast pulses, Phys. Rev. Lett. **88**, 207902 (2002).
[50] A. Vezvaee, V. Tripathi, D. Kowsari, E. Levenson-Falk, and D. A. Lidar, Virtual Z gates and symmetric gate compilation, arXiv:2407.14782 [quant-ph].
[51] A. Seif, H. Liao, V. Tripathi, K. Krsulich, M. Malekakhlagh, M. Amico, P. Jurcevic, and A. Javadi-Abhari, "Suppressing correlated noise in quantum computers via context-aware compiling," in *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)* (IEEE, Buenos Aires, Argentina, 2024), pp. 310–324.
[52] T. Proctor, K. Rudinger, K. Young, E. Nielsen, and R. Blume-Kohout, Measuring the capabilities of quantum computers, Nat. Phys. **18**, 75 (2022).
[53] S. Aaronson and D. Gottesman, Improved simulation of stabilizer circuits, Phys. Rev. A **70**, 052328 (2004).
[54] E. Polak, *Optimization: Algorithms and Consistent Approximations* (Springer Science & Business Media, New York, N.Y, 2012), Vol. 124.
[55] D. J. Egger, C. Capecci, B. Pokharel, P. K. Barkoutsos, L. E. Fischer, L. Guidoni, and I. Tavernelli, Pulse variational quantum eigensolver on cross-resonance-based hardware, Phys. Rev. Res. **5**, 033159 (2023).
[56] A. Rahman, D. J. Egger, and C. Arenz, Learning how to dynamically decouple by optimizing rotational gates, Phys. Rev. Appl. **22**, 054074 (2024).
[57] S. Niu and A. Todri-Sanial, Analyzing strategies for dynamical decoupling insertion on IBM quantum computer, arXiv preprint arXiv:2204.14251.

[58] C. Reeves and J. E. Rowe, *Genetic Algorithms: Principles and Perspectives: a Guide to GA Theory* (Springer Science & Business Media, New York, N.Y, 2002), Vol. 20.

[59] M. Mitchell, *An Introduction to Genetic Algorithms* (MIT Press, Cambridge, 1999).

[60] A. F. Brown and D. A. Lidar, Efficient chromatic-number-based multi-qubit decoherence and crosstalk suppression, arXiv:2406.13901 [quant-ph].

[61] Here I_m and I_p both act as the identity element, although I_m is included for purposes of group structure.

[62] To impose the condition that such allocations of identity operations indeed lead to distinct staggerings, the DD sequence associated with the first color in the strategy has the pulse sequence placed uniformly between the operations defining the start and end of the idle period, while the second receives an asymmetric sequence with a pulse at the earliest time one can be applied in the idle period and the third receives an asymmetric sequence with a pulse at the latest time one can be applied in the idle period.

[63] C. Chamberland, G. Zhu, T. J. Yoder, J. B. Hertzberg, and A. W. Cross, Topological and subsystem codes on low-degree graphs with flag qubits, Phys. Rev. X **10**, 011022 (2020).

[64] E. Bernstein and U. Vazirani, in *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing* (Association for Computing Machinery, New York, NY, USA, 1993), pp. 11–20.

[65] T. Lubinski, S. Johri, P. Varosy, J. Coleman, L. Zhao, J. Necaise, C. H. Baldwin, K. Mayer, and T. Proctor, Application-oriented performance benchmarks for quantum computing, IEEE Trans. Quantum Eng. **4**, 1 (2023).

[66] S. Fallek, C. Herold, B. McMahon, K. Maller, K. Brown, and J. Amini, Transport implementation of the Bernstein–Vazirani algorithm with ion qubits, New J. Phys. **18**, 083030 (2016).

[67] K. Wright, K. M. Beck, S. Debnath, J. Amini, Y. Nam, N. Grzesiak, J. Chen, N. Pisenti, M. Chmielewski, C. Collins, *et al.*, Benchmarking an 11-qubit quantum computer, Nat. Commun. **10**, 5464 (2019).

[68] The Qiskit Research developers and contributors, Qiskit Research, 2023.

[69] P. S. Mundada, A. Barbosa, S. Maity, Y. Wang, T. Merkh, T. Stace, F. Nielson, A. R. Carvalho, M. Hush, M. J. Biercuk, *et al.*, Experimental benchmarking of an automated deterministic error-suppression workflow for quantum algorithms, Phys. Rev. Appl. **20**, 024034 (2023).

[70] D. A. Lidar and T. A. Brun, *Quantum Error Correction* (Cambridge University Press, Cambridge, England, 2013).

[71] D. M. Greenberger, M. A. Horne, and A. Zeilinger, in *Bell's Theorem, Quantum Theory and Conceptions of the Universe*, edited by Menas Kafatos (Springer Netherlands, Dordrecht, 1989) pp. 69–72.

[72] M. Hillery, V. Bužek, and A. Berthiaume, Quantum secret sharing, Phys. Rev. A **59**, 1829 (1999).

[73] S. Omkar, S.-H. Lee, Y. S. Teo, S.-W. Lee, and H. Jeong, All-photonic architecture for scalable quantum computing with Greenberger-Horne-Zeilinger states, PRX Quantum **3**, 030309 (2022).

[74] B. Pankovich, A. Kan, K. H. Wan, M. Ostmann, A. Neville, S. Omkar, A. Sohbi, and K. Brádler, High-photon-loss threshold quantum computing using GHZ-state measurements, Phys. Rev. Lett. **133**, 050604 (2024).

[75] K. Chen and H.-K. Lo, Multi-partite quantum cryptographic protocols with noisy GHZ states, arXiv:quant-ph/0404133 [quant-ph].

[76] C. Song, K. Xu, H. Li, Y.-R. Zhang, X. Zhang, W. Liu, Q. Guo, Z. Wang, W. Ren, J. Hao, H. Feng, H. Fan, D. Zheng, D.-W. Wang, H. Wang, and S.-Y. Zhu, Generation of multicomponent atomic Schrödinger cat states of up to 20 qubits, Science **365**, 574 (2019).

[77] E. H. Chen, G.-Y. Zhu, R. Verresen, A. Seif, E. Bäumer, D. Layden, N. Tantivasadakarn, G. Zhu, S. Sheldon, A. Vishwanath, S. Trebst, and A. Kandala, Nishimori transition across the error threshold for constant-depth quantum circuits, Nat. Phys. **21**, 161 (2025).

[78] E. Magesan, J. M. Gambetta, and J. Emerson, Scalable and robust randomized benchmarking of quantum processes, Phys. Rev. Lett. **106**, 180504 (2011).

[79] E. Magesan, J. M. Gambetta, and J. Emerson, Characterizing quantum gates via randomized benchmarking, Phys. Rev. A **85**, 042311 (2012).

[80] T. Proctor, S. Seritan, K. Rudinger, E. Nielsen, R. Blume-Kohout, and K. Young, Scalable randomized benchmarking of quantum computers using mirror circuits, Phys. Rev. Lett. **129**, 150502 (2022).

[81] J. Hines, M. Lu, R. K. Naik, A. Hashim, J. Ville, B. Mitchell, J. M. Kriekebaum, D. I. Santiago, S. Seritan, E. Nielsen, R. Blume-Kohout, K. Young, I. Siddiqi, B. Whaley, and T. Proctor, Demonstrating scalable randomized benchmarking of universal gate sets, Phys. Rev. X **13**, 041030 (2023).

[82] D. C. McKay, I. Hincks, E. J. Pritchett, M. Carroll, L. C. G. Govia, and S. T. Merkel, Benchmarking quantum processor performance at scale, arXiv:2311.05933.

[83] M. Amico, H. Zhang, P. Jurcevic, L. S. Bishop, P. Nation, A. Wack, and D. C. McKay, in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, Bellevue, Washington, USA, 2023), Vol. 01, pp. 692–702.

[84] M. Malekakhlagh, E. Magesan, and D. C. McKay, First-principles analysis of cross-resonance gate operation, Phys. Rev. A **102**, 042605 (2020).

[85] L. F. Santos and L. Viola, Enhanced convergence and robust performance of randomized dynamical decoupling, Phys. Rev. Lett. **97**, 150501 (2006).

[86] https://github.com/qiskit-community/gadd.

[87] L. K. Grover, Quantum mechanics helps in searching for a needle in a haystack, Phys. Rev. Lett. **79**, 325 (1997).

[88] Qiskit Contributors, Qiskit: An open-source framework for quantum computing, (2023).

[89] N. Kanazawa, D. J. Egger, Y. Ben-Haim, H. Zhang, W. E. Shanks, G. Aleksandrowicz, and C. J. Wood, Qiskit Experiments: A Python package to characterize and calibrate quantum computers, J. Open Source Softw. **8**, 5329 (2023).

[90] B. Johnson, in *APS March Meeting Abstracts* (American Physical Society, Chicago, IL, USA, 2022), Vol. 2022, pp. T28–002.