

# Dynamical decoupling for superconducting qubits: a performance survey

Nic Ezzell,<sup>1,2,\*</sup> Bibek Pokharel,<sup>1,3,†</sup> Lina Tewala,<sup>4,5,‡</sup> Gregory Quiroz,<sup>4,6,§</sup> and Daniel A. Lidar<sup>1,3,7,8,¶</sup>

<sup>1</sup>*Department of Physics and Astronomy, University of Southern California, Los Angeles, CA, 90089, USA*

<sup>2</sup>*Center for Quantum Information Science & Technology,  
University of Southern California, Los Angeles, CA, 90089, USA*

<sup>3</sup>*Center for Quantum Information Science & Technology,  
University of Southern California, Los Angeles, CA 90089, USA*

<sup>4</sup>*Johns Hopkins University Applied Physics Laboratory, Laurel, MD, 20723, USA*

<sup>5</sup>*Thomas C. Jenkins Department of Biophysics, Johns Hopkins University Baltimore, MD, 21218, USA*

<sup>6</sup>*William H. Miller III Department of Physics & Astronomy,  
Johns Hopkins University, Baltimore, Maryland 21218, USA*

<sup>7</sup>*Department of Electrical Engineering, University of Southern California, Los Angeles, CA, 90089, USA*

<sup>8</sup>*Department of Chemistry, University of Southern California, Los Angeles, CA, 90089, USA*

(Dated: July 11, 2022)

Dynamical Decoupling (DD) is perhaps the simplest and least resource-intensive error suppression strategy for improving quantum computer performance. Here we report on a large-scale survey of the performance of 60 different DD sequences from 10 families, including basic as well as advanced sequences with high order error cancellation properties and built-in robustness. The survey is performed using three different superconducting-qubit IBMQ devices, with the goal of assessing the relative performance of the different sequences in the setting of arbitrary quantum state preservation. We find that the high-order universally robust (UR) and quadratic DD (QDD) sequences generally outperform all other sequences across devices and pulse interval settings. Surprisingly, we find that DD performance for basic sequences such as CPMG and XY4 can be made to nearly match that of UR and QDD by optimizing the pulse interval, with the optimal interval being substantially larger than the minimum interval possible on each device.

In the pre-fault-tolerance era, quantum computing research has two main near-term goals: to demonstrate the promise of quantum computers via proof of principle calculations, and to understand how error correction and other noise mitigations methods can pave a path towards building eventual fault-tolerant quantum computers. The last decade has seen the rise of multiple cloud-based quantum computing platforms that allow a community of researchers to experimentally test error suppression and correction techniques [1–11]. Error suppression using dynamical decoupling (DD) [12–16] is among the earliest methods to have been experimentally demonstrated, using experimental platforms such as trapped ions [17, 18], photonic qubits [19], electron paramagnetic resonance [20], nuclear magnetic resonance (NMR) [21–23], trapped atoms [24] and nitrogen vacancies in diamond [25]. It is known that DD can be used to improve the fidelity of quantum computation both without [26–33] and with quantum error correction [34, 35]. Several recent experimental studies have shown that DD can unequivocally improve the performance of superconducting-qubit based devices [36–39].

In this work, we systematically compare a suite of known and increasingly elaborate DD sequences developed over the past two decades (see Table I for a complete list and Ref. [40] for a detailed review). These DD sequences reflect a growing understanding of how to build features that suppress noise to increasingly higher order and with greater robustness to pulse

imperfections. Our goal is to study the efficacy of the older and the more recent advanced sequences on currently available quantum computers. To this end, we implement these sequences on three different IBM Quantum Experience (IBMQE) transmon qubit-based platforms: ibmq\_armonk (Armonk), ibmq\_bogota (Bogota), and ibmq\_jakarta (Jakarta). We rely on the open-pulse functionality [41] of IBMQE, which enables us to precisely control the pulses and their timing. The circuit-level implementation of the various sequences can be suboptimal, as we detail in the Appendix.

We assess these DD sequences for their ability to preserve an arbitrary single-qubit state. By and large, we find that all DD sequences outperform the “unprotected” evolution (without DD). The higher-order DD sequences, like concatenated DD (CDD [42]), Uhrig DD (UDD [43]), quadratic DD (QDD [44]), nested UDD (NUDD [45]) and universally robust (UR [46]), perform consistently well across devices and pulse placement settings. While these more elaborate sequences are statistically better than the traditional sequences such as Hahn echo [47], Carr-Purcell-Meiboom-Gill (CPMG), and XY4 [48] for short pulse intervals, their advantage diminishes with sparser pulse placement. As both systematic and random errors, e.g., due to finite pulse-width and limited control, are reduced, advanced sequences will likely provide further performance improvements. Overall, our study indicates that they can be viewed as the preferred choice over their traditional counterparts.

The structure of this paper is as follows. In Section I we review the pertinent DD background and describe the various pulse sequences we tested. In Section II, we detail the experimental setup, the nuances of DD sequence implementation, and the chosen success metrics. We describe the results

\* nezzell@usc.edu

† pokharel@usc.edu

‡ lina.tewala@emory.edu

§ gregory.quiroz@jhuapl.edu

¶ lidar@usc.edu

and what we learned about the sequences and devices in Section III. A summary of results and possible future research directions are provided in Section IV.

## I. DYNAMICAL DECOUPLING BACKGROUND

For completeness, we first provide a brief review of DD. In this section we focus on a small subset of all sequences studied in this work, primarily to introduce key concepts and notation. The details of all the other sequences are provided in Appendix A. The reader who is already an expert in the theory may wish to skim this section to become familiar with our notation.

### A. DD with perfect pulses

Consider a time-independent Hamiltonian

$$H = H_S + H_B + H_{SB}, \quad (1)$$

where  $H_S$  and  $H_B$  contains terms that act, respectively, only on the system or the bath, and  $H_{SB}$  contains the system-bath interactions. We write  $H_S = H_S^0 + H_S^1$ , where  $H_S^1$  represents an undesired, always on term (e.g., due to cross-talk), so that

$$H_{\text{err}} = H_S^1 + H_{SB} \quad (2)$$

represents the “error Hamiltonian” we wish to remove using DD.  $H_S^0$  contains all the terms we wish to keep. The corresponding *free* unitary evolution for duration  $\tau$  is given by

$$f_\tau \equiv U(\tau) = \exp(-i\tau H). \quad (3)$$

DD is generated by an additional, time-dependent control Hamiltonian  $H_c(t)$  acting purely on the system, so that the total Hamiltonian is

$$H(t) = H_S^0 + H_{\text{err}} + H_B + H_c(t). \quad (4)$$

An “ideal”, or “perfect” pulse sequence is generated by a control Hamiltonian that is a sum of error-free, instantaneous Hamiltonians  $\{\Omega_0 H_{P_k}\}_{k=1}^n$  that generate the pulses at corresponding intervals  $\{\tau_k\}_{k=1}^n$ :

$$\hat{H}_c(t) = \Omega_0 \sum_{k=1}^n \delta(t - t_k) H_{P_k}, \quad t_k = \sum_{j=1}^k \tau_j, \quad (5)$$

where we use the hat notation to denote ideal conditions and let  $\Omega_0$  have units of energy. Choosing  $\Omega_0$  such that  $\Omega_0 \Delta = \pi/2$ , where  $\Delta$  is the “width” of the Dirac-delta function (this is made rigorous when we account for pulse width in Section IB 1 below), this gives rise to instantaneous unitaries or pulses

$$\hat{P}_k = e^{-i\frac{\pi}{2} H_{P_k}}, \quad (6)$$

so that the total evolution is:

$$\tilde{U}(T) = f_{\tau_n} \hat{P}_n \cdots f_{\tau_2} \hat{P}_2 f_{\tau_1} \hat{P}_1, \quad (7)$$

where  $T \equiv t_n = \sum_{j=1}^n \tau_j$  is the total sequence time. The unitary  $\tilde{U}(T) = U_0(T)B(T)$  can be decomposed into the desired error-free evolution  $U_0(T) = \exp(-iT H_S^0) \otimes I_B$  and the unitary error  $B(T)$ . Ideally,  $B(T) = I_S \otimes e^{-iT \tilde{B}}$ , where  $\tilde{B}$  is an arbitrary Hermitian bath operator. Hence, by applying  $N$  repetitions of an ideal DD sequence of duration  $T$ , the system stroboscopically decouples from the bath at uniform intervals  $T_j = jT$  for  $j = 1, \dots, N$ . In reality, we only achieve approximate decoupling, so that  $B(T) = I_S \otimes e^{-iT \tilde{B}} + \text{err}$ , and the history of DD design is motivated by making the error term as small as possible under different and increasingly more realistic physical scenarios.

#### 1. First order protection

Historically, the first observation of stroboscopic decoupling came from nuclear magnetic resonance (NMR) spin echoes observed by Erwin Hahn in 1950 [47] with a single  $X$  pulse.<sup>1</sup> Several years later, Carr & Purcell [49] and Meiboom & Gill [50] independently proposed the improved CPMG sequence with two  $X$  pulses. In theory, both sequences are only capable of partial decoupling in the ideal pulse limit. In particular,  $B(T) \approx I_S \otimes e^{-iT \tilde{B}}$  only for states near  $|\pm\rangle = (|0\rangle \pm |1\rangle)/\sqrt{2}$  (where  $|0\rangle$  and  $|1\rangle$  are the  $+1$  and  $-1$  eigenstates of  $\sigma^z$ , respectively), as we explain below. Nearly four decades after Hahn’s work, Maudsley proposed the XY4 sequence [48], which is *universal* since  $B(T) \approx I_S \otimes e^{-iT \tilde{B}}$  on the full Hilbert space, which means all states are equally protected. Equivalently, universality means that arbitrary single-qubit interactions with the bath are decoupled to first order in  $\tau$ .

To make this discussion more precise, we first write  $H_B + H_{SB}$  in a generic way for a single qubit:

$$\bar{H} \equiv H_B + H_{SB} = \sum_{\alpha=0}^3 \gamma_\alpha \sigma^\alpha \otimes B^\alpha, \quad (8)$$

where  $B^\alpha$  are bath terms and  $\sigma^{(0)} = I$ . Since distinct Pauli operators anti-commute, i.e.,  $\{\sigma_i, \sigma_j\} = 2I\delta_{ij}$ , then for  $k \neq 0$ ,

$$\sigma_k \bar{H} \sigma_k = - \sum_{\alpha \neq k} \gamma_\alpha \sigma^\alpha \otimes B^\alpha + \gamma_k \sigma_k \otimes B_k. \quad (9)$$

The minus sign is an effective time-reversal of the terms that anticommute with  $\sigma_k$ . In the ideal pulse limit, this is enough to show that “pure- $X$ ” defined as

$$PX \equiv X - f_\tau - X - f_\tau, \quad (10)$$

induces an effective error Hamiltonian

$$H_{PX}^{\text{eff}} = \gamma_x \sigma^x \otimes B^x + I_S \otimes \tilde{B} + \mathcal{O}(\tau^2) \quad (11)$$

<sup>1</sup> We use  $X = \sigma^x$  interchangeably, and likewise for  $Y$  and  $Z$ , where  $\sigma^\alpha$  denotes the  $\alpha$ ’th Pauli matrix, with  $\alpha \in \{x, y, z\}$ . See Appendix A for a precise definition of all sequences.

every  $2\tau$ . Note that CPMG is defined similarly:

$$\text{CPMG} \equiv f_{\tau/2} - X - f_{\tau} - X - f_{\tau/2}, \quad (12)$$

which is just a symmetrized placement of the pulse intervals; see Section IC. PX and CPMG have the same properties in the ideal pulse limit, but we choose to begin with PX for simplicity of presentation. Intuitively, the middle  $X - f_{\tau} - X$  is a time-reversed evolution of the  $\sigma^{(y,z)}$  terms, followed by a forward evolution, which cancel to first order in  $\tau$  using the Zassenhaus formula [51],  $\exp\{t\tau(A+B)\} = e^{t\tau A} e^{t\tau B} + \mathcal{O}(\tau^2)$ , an expansion that is closely related to the familiar Baker-Campbell-Hausdorff (BCH) formula. The undesired noise term  $\gamma_x \sigma^x \otimes B^x$  does not decohere  $|\pm\rangle$ , but all other states are subject to bit-flip noise in the absence of suppression. By adding a second rotation around  $y$ , the XY4 sequence,

$$\text{XY4} \equiv Y - f_{\tau} - X - f_{\tau} - Y - f_{\tau} - X - f_{\tau} \quad (13)$$

cancels the remaining  $\sigma^x$  term and achieves universal (first order) decoupling at time  $4\tau$ :

$$H_{\text{XY4}}^{\text{eff}} = I_S \otimes \tilde{B} + \mathcal{O}(\tau^2). \quad (14)$$

Practically, this means that all single-qubit states are equally protected to first order. These results can be generalized by viewing DD as a symmetrization procedure [52], with an intuitive geometrical interpretation wherein the pulses replace the original error Hamiltonian by a sequence of Hamiltonians that are arranged symmetrically so that their average cancels out [53].

## 2. Higher order protection

While the XY4 sequence is universal for qubits, it only provides first-order protection. A great deal of effort has been invested in developing DD sequences that provide higher order protection. We start with concatenated dynamical decoupling, or CDD<sub>n</sub> [42]. CDD<sub>n</sub> is an  $n^{\text{th}}$ -order recursion of some base sequence. For example, CDD<sub>1</sub>  $\equiv$  XY4 is the base case, and

$$\text{CDD}_n \equiv \text{XY4}([\text{CDD}_{n-1}]) \quad (15a)$$

$$= Y - [\text{CDD}_{n-1}] - X - [\text{CDD}_{n-1}] - Y - [\text{CDD}_{n-1}] - X - [\text{CDD}_{n-1}], \quad (15b)$$

which is just the definition of XY4 in Eq. (13) with every  $f_{\tau}$  replaced by CDD<sub>n-1</sub>. This recursive structure leads to an improved error term  $\mathcal{O}(\tau^{n+1})$  provided  $\tau$  is “small enough.” To make this point precise, we must define a measure of error under DD. Following Ref. [34], one useful way to do this is to separate the “good” and “bad” parts of the joint system-bath evolution, i.e., to split  $\tilde{U}(T)$  [Eq. (7)] as

$$\tilde{U}(T) = \mathcal{G} + \mathcal{B}, \quad (16)$$

where  $\mathcal{G} = U_0(T) \otimes B'(T)$ , and where – as above –  $U_0(T)$  is the ideal operation that would be applied to the system in the absence of noise, and  $B'(T)$  is a unitary transformation acting

purely on the bath. The operator  $\mathcal{B}$  is the “bad” part, i.e., the deviation of  $\tilde{U}(T)$  from the ideal operation. The error measure is then<sup>2</sup>

$$\eta_{\text{DD}} = \|B'(T)\|. \quad (17)$$

Put simply,  $\eta_{\text{DD}}$  measures how far the DD-protected evolution  $\tilde{U}(T)$  is from the ideal evolution  $\mathcal{G}$ . With this error measure established, we can bound the performance of various DD sequences in terms of the relevant energy scales:

$$\beta \equiv \|H_B\|, \quad J \equiv \|H_{SB}\|, \quad \varepsilon \equiv \beta + J. \quad (18)$$

Using these definitions, we can replace the coarse  $\mathcal{O}$  estimates with rigorous upper bounds on  $\eta_{\text{DD}}$ . In particular, as shown in Ref. [34],

$$\eta_{\text{XY4}} = (4J\tau) \left[ \frac{1}{2}(4\varepsilon\tau) + \frac{2}{9}(4\varepsilon\tau)^2 \right] + \mathcal{O}(\tau^3) \quad (19a)$$

$$\eta_{\text{CDD}_n} = 4^{n(n+3)/2} (c\varepsilon\tau)^n (J\tau) + \mathcal{O}(\tau^{n+2}), \quad (19b)$$

where  $c$  is a constant of order 1. This more careful analysis implies that (1)  $\varepsilon\tau \lesssim 1$  is sufficient for XY4 to provide error suppression, and (2) CDD<sub>n</sub> has an optimal concatenation level induced by the competition between taking longer (the bad  $\sim 4^{n^2}$  scaling) and more error suppression [the good  $(c\varepsilon\tau)^n$  scaling]. The corresponding optimal concatenation level is

$$n_{\text{opt}} = \lfloor \log_4(1/\bar{c}\varepsilon\tau) - 1 \rfloor, \quad (20)$$

where  $\lfloor \cdot \rfloor$  is the floor function and  $\bar{c}$  is another constant of order 1 (defined in Eq. (165) of Ref. [34]). That such a saturation in performance should occur is fairly intuitive. By adding more layers of recursion, we suppress noise that was unsuppressed before. However, at the same time, we introduce more periods of free evolution  $f_{\tau}$  which cumulatively add up to more noise. At some point, the noise wins since there is no active noise removal in an open loop procedure such as DD.

Though CDD<sub>n</sub> derived from recursive symmetrization allows for  $\tau^n$  order suppression, it employs  $\sim 4^n$  pulses. One may ask whether a shorter sequence could achieve the same goal. The answer is provided by the Uhrig DD (UDD) sequence [43]. The idea is to find which DD sequence acts as an optimal filter function on the noise-spectral density of the bath while relaxing the constraint of uniform pulse intervals [43, 54–56]. For a brief overview, we first assume that a qubit state decoheres as  $e^{-\chi(t)}$ . For a given noise spectral density  $S(\omega)$ ,

$$\chi(t) = \frac{2}{\pi} \int_0^\infty \frac{S(\omega)}{\omega^2} F(\omega t) d\omega, \quad (21)$$

<sup>2</sup> We use the sup operator norm (the largest singular value of  $A$ ):  $\|A\| \equiv \sup_{\{|\nu\rangle\}} \frac{\|A|\nu\rangle\|}{\| |\nu\rangle \|} = \sup_{\{|\nu\rangle \text{ s.t. } \|\nu\|=1\}} \|A|\nu\rangle\|$ .

where the frequency response of the system to DD is captured by the filter function  $F(\omega t)$ . For example, for  $n$  ideal  $\pi$  pulses executed at times  $\{t_j\}$  [57],

$$F_n(\omega\tau) = \left| 1 + (-1)^{n+1} e^{i\omega\tau} + 2 \sum_{j=1}^n (-1)^j e^{i\omega t_j} \right|^2, \quad (22)$$

which can be substituted into Eq. (21) and optimized for  $\{t_j\}$ ; the result is UDD [43]. For a desired total evolution  $T$ , the solution (and definition of UDD) is simply to place  $\pi$  pulses with *non-uniform* pulse intervals,

$$t_j = T \sin\left(\frac{j\pi}{2(n+1)}\right)^2. \quad (23)$$

When we use  $n$   $X$ -type pulses in particular, we obtain  $\text{UDD}x_n$ . It turns out that  $\text{UDD}x_n$  achieves  $\mathcal{O}(\tau^n)$  suppression for states near  $|\pm\rangle$  using only  $n$  pulses, and this is the minimum number of  $\pi$  pulses needed [43, 58]. It is in this sense that UDD is provably optimal. However, it is important to note that this assumes that the total sequence time is fixed; only in this case can the optimal sequence be used to make the distance between the protected and unperturbed qubit states arbitrarily small in the number of applied pulses. On the other hand, if the minimum pulse interval is fixed and the total sequence time is allowed to scale with the number of pulses, then – as in CDD – longer sequences need not always be advantageous [59].

UDD can be improved from a single axis sequence to the universal *quadratic* DD sequence (QDD) [44, 58, 60] using recursive design principles similar to those that lead to XY4 and eventually  $\text{CDD}_n$  from PX. Namely, to achieve universal decoupling, we use a recursive embedding of a  $\text{UDD}y_m$  sequence into a  $\text{UDD}x_n$  sequence. Each  $X$  pulse in  $\text{UDD}x_n$  is separated by a free evolution period  $f_{t_{j+1}-t_j}$  which can be filled with a  $\text{UDD}y_m$  sequence. Hence, we can achieve  $\min\{\tau^n, \tau^m\}$  universal decoupling, and when  $m = n$ , we obtain universal order  $\tau^n$  decoupling using only  $n^2$  pulses instead of the  $\sim 4^n$  in  $\text{CDD}_n$ . This is nearly optimal [44], and an exponential improvement over  $\text{CDD}_n$ . When  $m \neq n$ , the exact decoupling properties are more complicated [58]. Similar comments as for  $\text{UDD}x_n$  regarding the difference between a fixed total sequence time  $T$  vs a fixed minimum pulse interval apply for QDD as well [61].

While QDD is universal and near-optimal for single-qubit decoherence, the ultimate recursive generalization is nested UDD (NUDD) [45], which applies for general multi-qubit decoherence, and whose universality and suppression properties have been proven and analyzed in a number of works [61–63]. In the simplest setting, suppression to  $N$ 'th order of general decoherence afflicting an  $m$ -qubit system requires  $(N+1)^{2m}$  pulses under NUDD.

## B. DD with imperfect pulses

So far, we have reviewed DD theory with ideal pulses. An ideal pulse is instantaneous and error-free, but in reality, finite

sequence	uniform interval	universal	Needs OpenPulse
Hahn Echo [47]	Y	N	N
PX/ CPMG [49, 64]	Y	N	N
XY4 [48]	Y	Y	Y/N
$\text{CDD}_n$ [42]	Y	Y	Y
EDD [65]	Y	Y & N	Y
$\text{RGA}_n$ [66]	Y	Y & N	Y
KDD [67]	Y	Y	Y
$\text{UR}_n$ [46]	Y	Y	Y
$\text{UDD}x_n$ [43]	N	N	N
$\text{QDD}_{n,m}$ [44]	N	Y	Y

TABLE I. Summary of the DD sequences surveyed in this work, along with the original references. A sequence has a uniform (pulse) interval provided  $\tau_i = \tau_j \forall i, j$  [see Eq. (7)] and is non-uniform otherwise. A sequence is universal (in theory) if it cancels an arbitrary  $H_{\text{err}}$  [Eq. (2)] to first order in the pulse interval, and practically this means it protects all states equally well. Otherwise, it only protects a subset of states (e.g., CPMG, which only protects  $|\pm\rangle$ ). In practice, this distinction is more subtle due to rotating frame effects, as discussed in Section ID. For those listed as both Y & N, such as  $\text{RGA}_n$ , we mean that not all sequences in the family are universal. For example,  $\text{RGA}_2$  is not universal but  $\text{RGA}_n$  for  $n \geq 4$  is universal. The last column lists whether our eventual implementation requires OpenPulse [68] or can be implemented faithfully just with the traditional circuit API [69, 70] (see Appendix B).

bandwidth constraints and control errors matter. Much of the work since CPMG has been concerned with (1) accounting for finite pulse width, (2) mitigating errors induced by finite width, and (3) mitigating systematic errors such as over- or under-rotations. We shall address these concerns in order.

### 1. Accounting for finite pulse width

During a finite width pulse,  $P_j$ , the effect of  $H_{\text{err}} + H_B$  cannot be ignored, so the analysis of Section IA needs to be modified correspondingly. Nevertheless, both the symmetrization and filter function approaches can be augmented to account for finite pulse width.

We may write a realistic DD sequence with  $\Delta$ -width pulses just as in Eq. (7), but with the ideal control Hamiltonian replaced by

$$\hat{H}_c(t) = \sum_k \Omega(t - \tau_k) H_{P_k}, \quad \int_{-\Delta/2}^{\Delta/2} \Omega(t) dt = \Omega_0 \Delta = \frac{\pi}{2}, \quad (24)$$

where  $\Omega(t)$  is sharply (but not infinitely) peaked at  $t = 0$  and vanishes for  $|t| > \Delta/2$ . The corresponding DD pulses are of the form

$$P_k \equiv \exp\left\{-i \int_{\tau_k - \Delta/2}^{\tau_k + \Delta/2} dt [\Omega(t - \tau_k) H_{P_k} + H_{\text{err}} + H_B]\right\}. \quad (25)$$

Note that the pulse intervals remain  $\tau_k$  as before, now denoting the peak-to-peak interval; the total sequence time therefore remains  $T = \sum_{j=k}^n \tau_k$ . The ideal pulse limit of Eq. (7) is obtained by taking the pulse width to zero, so that  $H_{\text{err}} + H_B$



can be ignored:

$$\hat{P}_k = \lim_{\Delta \rightarrow 0} P_k = e^{-i\frac{\pi}{2}H_{P_k}}, \quad \lim_{\Delta \rightarrow 0} \Omega(t) = \Omega_0 \delta(t). \quad (26)$$

We can then recover a result similar to Eq. (9) by entering the toggling frame with respect to the control Hamiltonian  $H_c(t)$  (see Appendix C), and computing  $\eta_{DD}$  with a Magnus expansion or Dyson series [34]. Though the analysis is involved, the final result is straightforward:  $\eta_{DD}$  picks up an additional dependence on the pulse width  $\Delta$ . For example, Eq. (19a) is modified to

$$\eta_{XY4}^{(\Delta)} = 4J\Delta + \eta_{XY4}, \quad (27)$$

which now has a linear dependence on  $\Delta$ . This new dependence is fairly generic, i.e., the previously discussed PX, XY44, CDD<sub>n</sub>, UDD<sub>n</sub>, and QDD<sub>n,m</sub> all have an error  $\eta$  with an additive  $\mathcal{O}(\Delta)$  dependence. Nevertheless, (1) DD is still effective provided  $J\Delta \ll 1$ , and (2) concatenation to order  $n$  is still effective provided the  $J\Delta$  dependence does not dominate the  $\varepsilon\tau^n$  dependence. For CDD<sub>n</sub> this amounts to an effective noise strength floor [34],

$$\eta_{CDD_n} \geq 16\Delta J, \quad (28)$$

which modifies the optimal concatenation level  $n_{opt}$ .

### 2. Mitigating errors induced by finite width

A natural question is to what extent we can suppress this first order  $\mathcal{O}(\Delta)$  dependence. One solution is Eulerian symmetrization,<sup>3</sup> which exhibits robustness to pulse-width errors [65, 71, 72]. For example, the palindromic sequence

$$\text{EDD} \equiv X f_\tau Y f_\tau X f_\tau Y f_\tau Y f_\tau X f_\tau Y f_\tau X f_\tau, \quad (29)$$

which is an example of Eulerian DD (EDD), has error term [34]

$$\eta_{\text{EDD}} = (8J\tau) \left[ \frac{1}{2}(8\varepsilon\tau) + \frac{2}{9}(8\varepsilon\tau)^2 + \mathcal{O}(\tau^3) \right], \quad (30)$$

which contains no first order  $\mathcal{O}(\Delta)$  term. Nevertheless, the constant factors are twice as large compared to XY4, and it turns out that EDD outperforms XY4 when  $\Delta/\tau \gtrsim 8\varepsilon\tau$  (see Fig. 9 in Ref. [34]).<sup>4</sup> The same Eulerian approach can be used to derive the pulse-width robust version of the Hahn echo and CPMG, which we refer to with a “super” prefix (derived from the “Eulerian supercycle” terminology of Ref. [72]):

$$\text{super-Hahn} \equiv X f_\tau \bar{X} f_\tau \quad (31a)$$

$$\text{super-CPMG} \equiv X f_\tau X f_\tau \bar{X} f_\tau \bar{X} f_\tau, \quad (31b)$$

where

$$\bar{P}_k \equiv \exp \left\{ -i \int_{\tau_k - \Delta/2}^{\tau_k + \Delta/2} dt [-\Omega(t - \tau_k) H_{P_k} + H_{\text{err}} + H_B] \right\} \quad (32)$$

[compare to Eq. (25)]. Intuitively, if  $X$  is a finite pulse that generates a rotation about the  $x$  axis of the Bloch sphere, then  $\bar{X}$  is (approximately) a rotation about the  $-x$  axis, i.e., with opposite orientation.

These robust sequences, coupled with concatenation, suggest that we can eliminate the effect of pulse width to arbitrary order  $\mathcal{O}(\Delta^n)$ ; up to certain caveats this indeed holds with concatenated dynamically corrected gates (CDCG) [73] (also see Ref. [74]).<sup>5</sup> However, this approach deviates significantly from the sequences consisting of only  $\pi$  rotations we have considered so far. To our knowledge, no strategy better than EDD exists for sequences consisting of only  $\pi$  pulses [71].

### 3. Mitigating systematic errors

In addition to finite width errors, real pulses are also subject to systematic errors. For example,  $\Omega(t)$  might be slightly miscalibrated, leading to a systematic over- or under-rotation, and any aforementioned gain might be lost due to the accumulation of these errors. A useful model of pulses subject to systematic errors is

$$P_j^r = \exp \left\{ \pm i \frac{\pi}{2} (1 + \varepsilon_r) \sigma^\alpha \right\} \quad (33)$$

for  $\alpha \in \{x, y, z\}$ . This represents instantaneous  $X, Y, Z$  and  $\bar{X}, \bar{Y}, \bar{Z}$  pulses subject to systematic over- or under-rotation by  $\varepsilon_r$ , also known as a *flip-angle error*.

Fortunately, even simple  $\pi$  pulses can mitigate systematic errors if rotation axes other than  $+x$  and  $+y$  are used. We consider three types of sequences: robust genetic algorithm (RGA) DD [66], Knill DD (KDD) [67, 75] and universally robust (UR) DD [46].

*a. RGA DD.*— The basic idea of RGA is as follows. A universal DD sequence should satisfy  $\prod_{k=1}^n P_k = I$  up to a global phase, but there is a great deal of freedom in what combination of pulses are used that satisfy this constraint. In Ref. [66], this freedom was exploited to find, by numerical optimization with genetic algorithms, a class of sequences robust to over- or under-rotations.

Subject to a generic single-qubit error Hamiltonian as in Eq. (8), optimal DD sequences were then found for a given number of pulses under different parameter regimes (i.e., relative magnitude of  $J, \beta$ , etc.). This numerical optimization “rediscovered” CPMG, XY4, and Eulerian DD as base sequences

<sup>3</sup> The terminology arises from Euler cycles traversed by the control unitary in the Cayley graph of the group generated by the DD pulses.

<sup>4</sup> To clarify their relationship, suppose we take the ideal  $\Delta \rightarrow 0$  limit. Here, XY4 is strictly better since EDD uses twice as many pulses (and therefore free periods) to accomplish the same  $\mathcal{O}(\tau)$  decoupling.

<sup>5</sup> Caveats include: (1) The analytical CDCG constructions given in Ref. [73] do not accommodate for the setting where always-on terms in the system’s Hamiltonian are needed for universal control (so that they cannot just be included in  $H_{\text{err}}$ ); (2) Control-induced (often multiplicative) noise can be simultaneously present along with bath-induced noise; multiplicative noise requires modifications to the formalism of Ref. [73].

with  $\mathcal{O}(\tau^2)$  errors. Higher order sequences were then found to be concatenations of the latter. For example,

$$\text{RGA}_{8c} \equiv \text{EDD} \quad (34a)$$

$$\text{RGA}_{64c} \equiv \text{RGA}_{8c}[\text{RGA}_{8c}]. \quad (34b)$$

A total of 12 RGA sequences were found in total; more details are given in Appendix A.

*b. KDD.*— The KDD sequence is similar in its goal to RGA because it mitigates systematic over or under-rotations. In design, however, it uses the principle of composite pulses [76–78]. The idea is to take a universal sequence such as XY4 and replace each pulse  $P_j$  with a composite series of pulses  $(CP)_j$  that have the same effect as  $P_j$  but remove pulse imperfections by self-averaging; for details, see Appendix A.

The KDD sequence is robust to  $\varepsilon_r$  errors [Eq. (33)]. For example, if  $\varepsilon_r = \pi/20$ , then by direct calculation,  $\|(\text{KDD})^{10} - I\| \approx 7 \times 10^{-7}$  whereas  $\|(\text{XY4})^{10} - I\| \approx 3 \times 10^{-2}$ , and in fact, KDD is robust up to  $\mathcal{O}(\varepsilon_r^5)$  [40, 46]. This robustness to over-rotations comes at the cost of 20 free evolution periods instead of 4, so we only expect KDD to work well in an  $\varepsilon_r$  dominated parameter regime. As a preview of the results we report in Section III, KDD is not among the top performing sequences. Hence it appears reasonable to conclude that our experiments are conducted in a regime which is *not*  $\varepsilon_r$  dominated.

*c. UR DD.*— An alternative approach to devise robust DD sequences is the  $\text{UR}_n$  DD family [46], developed for a semiclassical noise model. In particular, the system Hamiltonian is modified by a random term, as opposed to including an explicit quantum bath and system-bath interaction as for the other DD sequences we consider in this work. The model is expressed using an arbitrary unitary pulse that includes a fixed systematic error  $\varepsilon_r$  as in Eq. (33), and reduces to a  $\pi$  pulse in the ideal case. As detailed in Appendix A, this leads to a family of sequences which give rise to an error scaling as  $\varepsilon_n \sim \varepsilon_r^{n/2}$  using  $n$  pulses.

These sequences recover some known results at low order:  $\text{UR}_2$  is CPMG, and  $\text{UR}_4$  is XY4. In other words, CPMG and XY4 are also robust to some pulse imperfections as they cancel flip-angle errors to a certain order. Moreover, by the same recursive arguments,  $\text{CDD}_n$  can achieve arbitrary flip-angle error suppression while achieving arbitrary  $\mathcal{O}(\tau^n)$  (up to saturation) protection. Still,  $\text{CDD}_n$  requires exponentially more pulses than  $\text{UR}_n$ , since  $\text{UR}_n$  is by design a semiclassical  $\mathcal{O}(\tau^2)$  sequence. Whether  $\text{UR}_n$  is also an  $\mathcal{O}(\tau^2)$  sequence for a fully quantum bath is an interesting open problem.

### C. Optimizing the pulse interval

Non-uniform pulse interval sequences such as UDD and QDD already demonstrate that introducing pulse intervals longer than the minimum possible ( $\tau_{\min}$ ) can be advantageous. In particular, such alterations can reduce spectral overlap between the filter function and bath spectral density. A longer pulse interval also results in pulses that are closer to the ideal limit of a small  $\Delta/\tau$  ratio when  $\Delta$  is fixed. Empirical studies

have also found evidence for better DD performance under longer pulse intervals [40, 56, 67].

We may distinguish two ways to optimize the pulse interval: an asymmetric or symmetric placement of additional delays. For example, the asymmetric and symmetric forms of XY4 we optimize take the form

$$\text{XY4}_a(d) \equiv Y f_d X f_d Y f_d X f_d \quad (35a)$$

$$\text{XY4}_s(d) \equiv f_{d/2} Y f_d X f_d Y f_d X f_{d/2}, \quad (35b)$$

where  $d$  sets the duration of the pulse interval. The asymmetric form here is consistent with how we defined XY4 in Eq. (13), and with the exception of CPMG we have tacitly defined every sequence so far in its asymmetric form for simplicity. The symmetric form is a simple alteration that makes the placement of pulse intervals symmetric about the mid-point of the sequence. For a generic sequence whose definition requires non-uniform pulse intervals like UDD, we can write the two forms as

$$\text{DD}_a(d) \equiv P_1 f_{\tau_1+d} P_2 f_{\tau_2+d} \dots P_n f_{\tau_n+d} \quad (36a)$$

$$\text{DD}_s(d) \equiv f_{d/2} P_1 f_{\tau_1+d} P_2 f_{\tau_2+d} \dots P_n f_{\tau_n+d/2}. \quad (36b)$$

Here, the symmetric nature of  $\text{DD}_s$  is harder to interpret. The key is to define  $\tilde{P}_j = P_1 f_{\tau_1}$  as the “effective pulse”. In this case, the delay-pulse motif is  $f_{d/2} \tilde{P}_j f_{d/2}$  for every pulse in the sequence which exhibits a reflection symmetry of the pulse interval about the center of the pulse. In the asymmetric version, it is  $\tilde{P}_j f_d$  instead. Note that  $\text{PX}_s(\tau) = \text{CPMG}$ , i.e., the symmetric form of the PX sequence is CPMG. As CPMG is a well-known sequence, henceforth we refer to the PX sequence as CPMG throughout our data and analysis regardless of symmetry.

### D. Superconducting hardware physics relevant to DD

So far, our account has been abstract and hardware agnostic. Since our experiments involve IBMQ superconducting hardware, we now provide a brief background relevant to the operation of DD in such devices. We closely follow Ref. [39], which derived the effective system Hamiltonian of transmon superconducting qubits from first principles and identified the importance of modeling DD performance within a frame rotating at the qubit drive frequency. It was found that DD is still effective with this added complication both in theory and experiment, and in practice DD performance can be modeled reasonably well by ideal pulses with a minimal pulse spacing of  $\tau_{\min} = \Delta$ . We now address these points in more detail.

The effective system Hamiltonian for two qubits (the generalization to  $n > 2$  is straightforward), is

$$H_S = -\frac{\omega_{q_1}}{2} Z_1 - \frac{\omega_{q_2}}{2} Z_2 + J Z_1 Z_2, \quad (37)$$

where the  $\omega_{q_i}$ ’s are qubit frequencies and  $J \neq 0$  is an undesired, always-on ZZ crosstalk. The appropriate frame for describing the superconducting qubit dynamics is the one co-rotating with the number operator  $\hat{N} = \sum_{k,l \in \{0,1\}} (k +$

$l)|kl\rangle\langle kl| = I - \frac{1}{2}(Z_1 + Z_2)$ . The unitary transformation into this frame is  $U(t) = e^{-i\omega_d \tilde{N}t}$ , where  $\omega_d$  is the drive frequency used to apply gates.<sup>6</sup> In this frame, the effective dynamics are given by the Hamiltonian

$$\tilde{H}(t) = \sum_{i=1}^2 \left( \frac{\omega_d - \omega_{qi}}{2} \right) Z_i + JZZ + \tilde{H}_{SB}(t) + H_B, \quad (38)$$

where  $\tilde{H}_{SB} = U^\dagger(t)H_{SB}U(t)$ . For DD to eliminate unwanted interactions, it must symmetrize  $JZZ$  and  $\tilde{H}_{SB}$ . The  $JZZ$  term is removed by applying an  $X$ -type DD to the first qubit (“the DD qubit”):  $X_1 Z_1 Z_2 X_1 = -Z_1 Z_2$ , so symmetrization still works as intended. However, the  $\tilde{H}_{SB}$  term is time-dependent in the rotating frame  $U(t)$ , which changes the analysis. First, the sign-flipping captured by Eq. (9) no longer holds due to the time-dependence of  $\tilde{H}_{SB}$ . Second, some terms in  $\tilde{H}_{err}$  self-average and nearly cancel even without applying DD:<sup>7</sup>

$$\{\sigma_1^x, \sigma_2^x, \sigma_1^y, \sigma_2^y, \sigma^x \sigma^z, \sigma^z \sigma^x, \sigma^y \sigma^z, \sigma^z \sigma^y\} \quad (39)$$

The remaining terms,

$$\{\sigma_2^z, \sigma^x \sigma^x, \sigma^x \sigma^y, \sigma^y \sigma^x, \sigma^y \sigma^y\}, \quad (40)$$

are not cancelled at all. Overall, this differs from expectation in that the two terms containing  $\sigma_1^y$  in Eq. (40) are not canceled, whereas in the  $\omega_d \rightarrow 0$  limit, all terms containing  $\sigma_1^y$  are fully canceled.

Somewhat surprisingly, a nominally universal sequence such as XY4, is no longer universal in the rotating frame, again due to the time-dependence acquired by  $\tilde{H}_{SB}$ . In particular, the only terms that perfectly cancel to  $\mathcal{O}(\tau)$  are the same terms,  $Z_1$  and  $ZZ$  as with CPMG. However, the list of terms that approximately cancels grows to include

$$\{\sigma^x \sigma^x, \sigma^x \sigma^y, \sigma^y \sigma^x, \sigma^y \sigma^y\}, \quad (41)$$

and when  $\tau$  is fine-tuned to an integer multiple of  $2\pi/\omega_d$ , then XY4 actually cancels all terms except, of course, terms involving  $I_1$ , which commute with the DD sequence.

Consequently, without fine-tuning  $\tau$ , we should expect CPMG and XY4 to behave similarly when the terms in Eq. (41) are not significant. Practically, this occurs when  $T_1 \gg T_2$  for the qubits coupled to the DD qubit [39]. However, when instead  $T_1 \lesssim T_2$  for coupled qubits, XY4 should prevail. In addition, the analysis in Ref. [39] was carried out under the assumption of ideal  $\pi$  pulses with  $\tau_{\min} = \Delta$ , and yet,

Device	ibmq_armonk	ibmq_bogota	ibmq_jakarta
# qubits	1	5	7
qubit used	q0	q2	q1
$T_1$ ( $\mu$ s)	$140 \pm 41$	$105 \pm 41$	$149 \pm 61$
$T_2$ ( $\mu$ s)	$227 \pm 71$	$145 \pm 63$	$21 \pm 3$
pulse duration (ns)	71.11	35.55	35.55

TABLE II. The three processors used in our experiments. The total number of qubits  $n$  varies from 1 to 7, but in all cases, we applied DD to just one qubit: number 2 for Bogota and 1 for Jakarta (see the insets of Fig. 3 for the connectivity graph of each device). The choice of the qubit used is motivated by the relative  $T_1$  and  $T_2$  times as well as these qubits having the highest connectivity. The  $T_1$  and  $T_2$  times are the averages of all reported values during data collection, along with the  $2\sigma$  sample standard deviation. Data was collected over roughly 20 distinct calibrations mostly between Aug. 9-25, 2021, for the Pauli experiment and Jan. 11-19, 2022, for the Haar-interval experiment.

the specific qualitative and quantitative predictions bore out in experiment. Hence, it is reasonable to model DD sequences on superconducting transmon qubits as

$$\text{DD}_{sc} \equiv \hat{P}_1 f_{\tau_1 + \Delta_1} \dots \hat{P}_n f_{\tau_n + \Delta_n}, \quad (42a)$$

where  $\hat{P}_j$  is once again an ideal pulse with zero width, and the free evolution periods have been incremented by  $\Delta_j$  – the width of the actual pulse  $P_j$ .

## II. METHODS

We performed our experiments on three IBM Quantum superconducting processors [79] with OpenPulse access [68, 70]: ibmq\_armonk, ibmq\_bogota, and ibmq\_jakarta. Key properties of these processors are summarized in Table II.

All our experiments follow the same basic structure, summarized in Fig. 1. Namely, we prepare a single-qubit initial state  $|\psi\rangle = U|0\rangle$ , apply  $N$  repetitions of a given DD sequence  $S$  lasting total time  $t$ , undo  $U$  by applying  $U^\dagger$ , and finally measure in the computational basis. Note that this is a single-qubit protocol. Even on multi-qubit devices, we intentionally only apply DD to the single qubit we intend to measure. This ensures that we cancel the unwanted  $ZZ$  term introduced by the aforementioned rotating frame [39]. The qubit used on each device is listed in Table II.

We empirically estimate the Uhlmann fidelity

$$f_e(t) = |\langle \psi | \rho_{\text{final}}(t) | \psi \rangle|^2 \quad (43)$$

with 95% confidence intervals by counting the number of 0’s returned out of 8192 shots and bootstrapping. The results we report below were obtained using OpenPulse [68], which allows for refined control over the exact waveforms sent to the chip instead of the coarser control that the standard Qiskit circuit API gives [69, 70]. Appendix B provides a detailed comparison between the two APIs, highlighting the significant advantage of using OpenPulse.

We utilize this simple procedure to perform two types of experiments which we refer to as Pauli and Haar, and explain

<sup>6</sup> This frame choice is motivated by the observation that in the IBMQ devices, the drive frequency is calibrated to be resonant with a particular eigenfrequency of the devices’ qubits, which depends on the state of the neighboring qubits. Transforming into a frame that rotates with one of these frequencies gets one as close as possible to describing the gates and pulses as static in the rotating frame.

<sup>7</sup> Intuitively, the  $Z_1 Z_2$  rotation at frequency  $\omega_d$  is already self-averaging the effects of noise around the  $z$ -axis, so in a sense, is acting like a DD sequence.

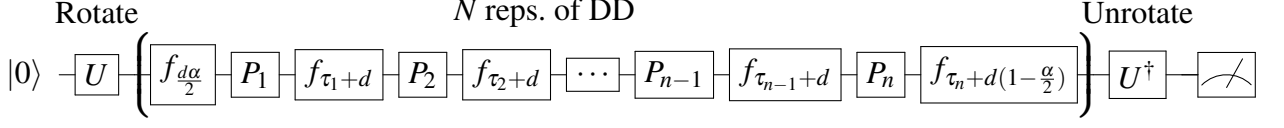


FIG. 1. The “quantum memory” circuit that underlies all of our experiments. By sampling the circuit using 8192 shots, we estimate the Uhlmann fidelity,  $f_e(t)$  in Eq. (43), between the initial prepared state and the final state under the DD sequence  $P_1 f_{\tau_1} P_2 f_{\tau_2} \dots P_n f_{\tau_n}$ . We have included an additional adjustable pulse interval  $f_d$ , where we set  $d = 0$  for the Pauli experiment (Section II A) and systematically vary  $d$  for the Haar experiment (Section II B). The choice of  $\alpha = 0$  (1) corresponds to an asymmetric (symmetric) placement of the additional delays.

in detail below. Briefly, the Pauli experiment probes the ability of DD to maintain the fidelity of the six Pauli states (the eigenstates of  $\{\sigma^x, \sigma^y, \sigma^z\}$ ) over long times, while the Haar experiments address this question for Haar-random states and short times.

### A. Pauli experiment for long times

For the Pauli experiment we keep the pulse interval  $\tau$  fixed to the smallest possible value allowed by the relevant device,  $\tau_{\min} = \Delta \approx 36\text{ns}$  (or  $\approx 71\text{ns}$ ), the width of the  $X$  and  $Y$  pulses (see Table II for specific  $\Delta$  values for each device). Practically, this corresponds to placing all the pulses as close to each other as possible, i.e., the peak-to-peak spacing between pulses is just  $\Delta$  (except for non-uniformly spaced sequences such as QDD). For ideal pulses and uniformly spaced sequences, this is expected to give the best performance [16, 80], so unless otherwise stated, all DD is implemented with minimal pulse spacing,  $\tau = \Delta$ .

Within this setting, we survey the capacity of each sequence to preserve the six Pauli eigenstates  $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle, |+i\rangle, |-i\rangle\}$  for long times, which we define as  $T \geq 75\mu\text{s}$ . In particular, we generate fidelity decay curves like those shown in Fig. 2(a) by incrementing the number of repetitions of the sequence,  $N$ , and thereby sampling  $f_e(t)$  [Eq. (43)] for increasingly longer times  $t$ . Using XY4 as an example, we apply  $P_1 P_2 \dots P_n = (XYXY)^N$  for different values of  $N$  while keeping the pulse interval fixed. After generating fidelity decay curves over 10 or more different calibration cycles across a week, we summarize their performance using a box-plot like that shown in Fig. 2(b). For the Pauli experiments, the box-plot bins the average normalized fidelity,

$$F(T) \equiv \frac{1}{f_e(0)} \langle f_e(t) \rangle_T = \frac{1}{T} \int_0^T dt \frac{f_e(t)}{f_e(0)}, \quad (44)$$

at time  $T$  computed using numerical integration with Hermite polynomial interpolation. Note that no DD is applied at  $f_e(0)$ , so we account for state preparation and measurement errors by normalizing. The same holds for Fig. 2(a).

We can estimate the best performing sequences for a given device by ranking them by the median performance from this data. In Fig. 2(b), for example, this leads to the fidelity ordering  $\text{UR}_{20} > \text{CPMG} > \text{XY4} > \text{free evolution}$  on Bogota, which agrees with the impression left by the decay profiles in Fig. 2(a) generated in a single run. We use  $F(T)$  because

fidelity profiles  $f_e(t)$  are generally oscillatory and noisy, so fitting  $f_e(t)$  to extract a decay constant (as was done in Ref. [36]) does not return reliable results across the many sequences and different devices we tested.

### B. Haar interval experiments

The Pauli experiment estimates how well a sequence preserves quantum memory for long times without requiring excessive data, but it leaves several open questions. Namely, (1) Does DD preserve quantum memory for an *arbitrary* state? (2) Is  $\tau = \tau_{\min}$  the best choice empirically? And, (3) how effective is DD for short times? In the Haar interval experiment, we address all of these questions. This setting – of short times and arbitrary states – is particularly relevant to improving quantum computations with DD [26, 30, 34, 81]. For example, DD pulses have been inserted into idle spaces to improve circuit performance [38, 82].

In contrast to the Pauli experiment, where we fixed the pulse delay  $d = 0$  and the symmetry  $\zeta = a$  (see Eq. (35a) and (35b)) and varied  $t$ , here we fix  $t = T$  and vary  $d$  and  $\zeta$ , writing  $f_e(d, \zeta; T)$ . Further, we now sample over a fixed set of 25 Haar random states instead of the 6 Pauli states. Note that we theoretically expect the empirical Haar-average over  $n$  states  $\mathbb{E}_{n-\text{Haar}}[f_e] \equiv \frac{1}{n} \sum_{i=1}^n f_e(\psi_i)$  for  $|\psi_i\rangle \sim \text{Haar}$  to converge to the true Haar-average  $\langle f_e \rangle_{\text{Haar}}$  for sufficiently large  $n$ . As shown by Fig. 2(c), 25 states are enough for a reasonable empirical convergence while keeping the total number of circuits to submit and run on IBMQ devices manageable in practice.

The Haar interval experimental procedure is now as follows. For a given DD sequence and time  $T$ , we sample  $f_e(d, \zeta; T)$  for  $\zeta \in \{a, s\}$  from  $d = 0$  to  $d = d_{\max}$  for 8 equally spaced values across 25 fixed Haar random states and 10 calibration cycles (250 data points for each  $d$  value). Here  $d = 0$  and  $d = d_{\max}$  correspond to the tightest and sparsest pulse placements, respectively. At  $d_{\max}$ , we consider only a single repetition of the sequence during the time window  $T$ . To make contact with DD in algorithms, we first consider a short-time limit  $T = T_{5\text{CNOT}} \approx 4\mu\text{s}$ , which is the amount of time it takes to apply 5 CNOTs on the DD-qubit. As shown by the example fidelity decay curves of Fig. 2(a), we expect similar results for  $T \lesssim 15\mu\text{s}$  before fidelity oscillations begin. To make contact with the Pauli experiment, we also consider a long-time limit of  $T = 75\mu\text{s}$ . Finally, to keep the number of experiments manageable, we only optimize the interval of (i) the best performing UR sequence, (ii) the best performing QDD sequence, and CPMG, XY4, and Free as constant references.



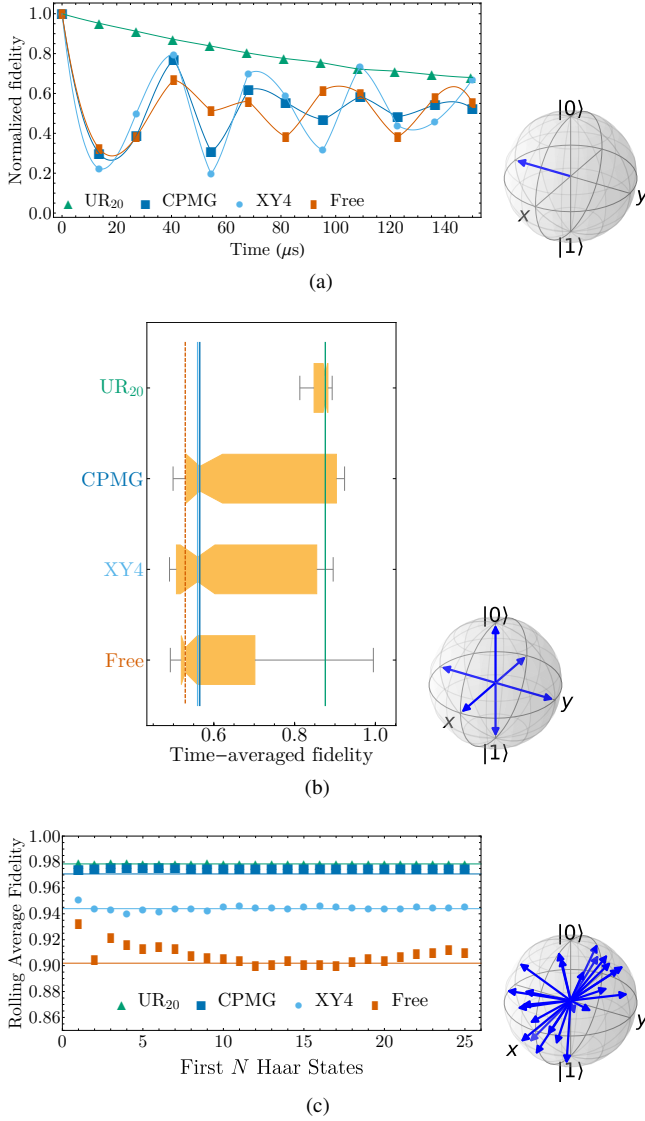


FIG. 2. Representative samples of our results for three DD sequences and free evolution. The Bloch sphere representation of the quantum states used for each plot is shown on the bottom right. (a) Normalized fidelity  $\frac{f_e(t)}{f_e(0)}$  under four DD sequences for the initial state  $|-i\rangle$  and a fixed calibration cycle on Bogota. (b) We summarize the result of many such fidelity decay curves using a box plot. Each box shows the max (right-most vertical black lines), inner-quartile range (the orange box), median (the skinny portion of each orange box), and minimum (left-most vertical black lines) time-averaged fidelity,  $F(T = 75\mu s)$  in Eq. (44), across the 6 Pauli states and 10 calibration cycles (for a total of 60 data points each) on Bogota. The vertical lines denote the performance of each sequence by its median, colored with the same color as the corresponding sequence. We use this exact type of box plot to summarize the Pauli experiment results. (c) We show average fidelity convergence as a function of number of Haar-random states. In particular, the horizontal lines represent  $\mathbb{E}_{100-\text{Haar}}[f_e(T = 3.27\mu s)]$  whereas each point represents  $\mathbb{E}_{N-\text{Haar}}[f_e(T = 3.27\mu s)]$  for increasing  $N$ , i.e., the rolling average fidelity. In all cases, we find that 25 states are sufficient for reasonable convergence as  $\mathbb{E}_{25-\text{Haar}}[f_e(T)]$  is within 1% of  $\mathbb{E}_{100-\text{Haar}}[f_e(T)]$ . The data shown is for Jakarta, but a similar result holds across all devices tested.

### III. RESULTS

We split our results into three subsections. The first two summarize the results of experiments aimed at preserving Pauli eigenstates (Section II A) and Haar-random states (Section II B). In the third subsection, we discuss how theoretical expectations about the saturation of  $\text{CDD}_n$  [80] and  $\text{UR}_n$  [46] compare to the experimental results.

#### A. The Pauli experiment result: DD works and advanced DD works even better

In Fig. 3, we summarize the results of the Pauli experiment. We rank the top 10 sequences for each device by median performance across the six Pauli states and 10 or more calibration cycles, followed by CPMG, XY4, and free evolution as standard references. As discussed in Section II A, the figure of merit is the normalized, time-averaged fidelity at  $75\mu s$  [see Eq. (44)] which is a long-time average.

The first major observation is that DD is better than free evolution, which is consistent with numerous previous DD studies. This is evidenced by free evolution (Free) being close to the bottom of the ranking for every device.

Secondly, advanced DD sequences outperform Free, CPMG, and XY4 (shown as dark-blue and light-blue vertical lines in Fig. 3). In particular, 29/30 top sequences across all three devices are advanced – the exception being XY4 on `ibmq_armonk`. These sequences perform so well that there is a 50% improvement in the median fidelity of these sequences (0.85-0.95) over Free (0.45-0.55). The best sequences also have a much smaller variance in performance, as evidenced by their smaller inter-quartile range in  $F$ . For example, on `ibmq_armonk` 75% of all experimental outcomes for  $F_{\text{UDDx25}}(75\mu s)$  fall between 0.9 and 0.95, whereas for Free, the same range is between 0.55 and 0.8. Similar comparisons show that advanced DD beats CPMG and XY4 for every device to varying degrees.

Among the top advanced sequences shown, 16/29  $\sim 55\%$  are UDD or QDD, which use non-uniform pulse intervals. On the one hand, the dominance of advanced DD strategies, especially UDD and QDD, is not surprising. After all, these sequences were designed to beat the simple sequences. On the other hand, as reviewed above, many confounding factors affect how well these sequences perform, such as finite pulse width errors and the effect of the rotating frame. It is remarkable that despite these factors, predictions made based on ideal pulses apply to real NISQ devices.

Finally, we comment more specifically on CPMG and XY4, as these are widely-used and well-known sequences. Generally, they do better than free evolution, which is consistent with previous results. On `ibmq_armonk` XY4 outperforms CPMG, which outperforms free evolution. On `ibmq_bogota`, both XY4 and CPMG perform comparably and marginally better than free evolution. Finally, On `ibmq_jakarta` XY4 is worse than CPMG – and even free evolution – but the median performance of CPMG is substantially better than that of free evolution. It is tempting to relate these results to the

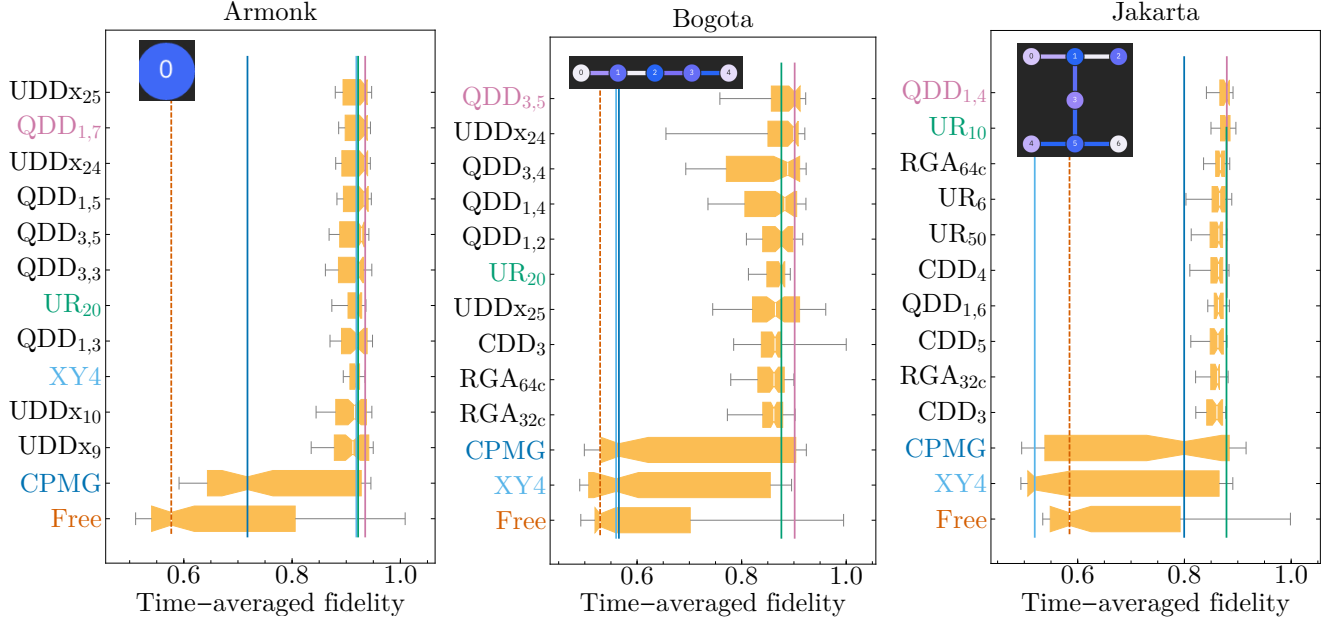


FIG. 3. A summary of the Pauli experiment results for all three devices. The top ten sequences are ranked from top to bottom by median average fidelity [Eq. (44)] for the listed time  $T = 75\mu s$ . Also displayed in all cases are CPMG and XY4, along with free evolution (Free). Colored vertical lines indicate the median fidelity of the correspondingly colored sequence (best UR, best QDD, CPMG, XY4, and free evolution). Thin white lines through the orange boxes indicate the median fidelities in all other cases. Otherwise the conventions of Fig. 2(b) apply. Two main observations emerge: (1) DD systematically outperforms free evolution as indicated by “Free” being at the bottom. The corresponding dot-dashed red vertical line denotes the median average fidelity of Free,  $F_{\text{Free}}(75\mu s)$ , which is below 0.6 on every device. (2) Advanced DD sequences, especially the UDD and QDD families, provide a substantial improvement over both CPMG and XY4. The best median performance of the top sequence is indicated by a solid green line,  $F_{\text{XY4}}(75\mu s)$  by a cyan line, and  $F_{\text{CPMG}}(75\mu s)$  by a dark blue line.

relative values of  $T_1$  and  $T_2$ , as per Table II, in the sense that CPMG is a single-axis (“pure-X”) type sequence [Eq. (10)] which does not suppress the system-bath  $\sigma^x$  coupling term responsible for  $T_1$  relaxation, while XY4 does. Nevertheless, a closer look at Fig. 3 shows such an explanation would be inconsistent with the fact that both single-axis and multi-axis sequences are among the top 10 performers for `ibmq_armonk` and `ibmq_bogota`.

The exception is `ibmq_jakarta`, for which there are no single-axis sequences in the top 10. This processor has a much smaller  $T_2$  than  $T_1$  (for `ibmq_armonk` and `ibmq_bogota`,  $T_1 > T_2$ ), so one might expect that a single-axis sequence such as UDD or CPMG would be among the top performers, but this is not the case. In the same vein, the top performing asymmetric QDD $_{n,m}$  sequences all have  $n < m$ , despite this opposite ordering of  $T_1$  and  $T_2$ . These results show that the backend values of  $T_1$  than  $T_2$  are not predictive of DD sequence performance.

### B. Haar Interval Experiment Results: DD works on arbitrary states and increasing the pulse interval can help substantially

We summarize the results for the Haar interval experiment in Fig. 4. Each plot corresponds to  $f_e(d, \zeta^*; T)$  as a function of  $d$ , the additional pulse interval spacing. For each device, we compare CPMG, XY4, the best robust sequence from

the Pauli Experiment, the best non-uniform sequence from the Pauli Experiment, and Free. The best robust and non-uniform sequences correspond to UR $_n$  and QDD $_{n,m}$  for each device. We only display the choice of  $\zeta$  with the better optimum, and the error bars correspond to the inner-quartile range across the 250 data points. These error bars are similar to the ones reported in the Pauli experiment.

#### 1. $d = 0$ : DD continues to outperform Free evolution also over Haar random states

The  $d = 0$  limit is identical to those in the Pauli Experiment, i.e., with the minimum possible pulse spacing. The advanced sequences, UR $_n$  and QDD $_{n,m}$ , outperform Free by a large margin for short times and by a moderate margin for long times. In particular, they have higher median fidelity, a smaller inter-quartile range than Free, and are consistently above the upper quartile in the short-time limit. UR $_n$  and QDD $_{n,m}$  are essentially indistinguishable in terms of their performance, except that UR $_n$  performs somewhat better in the long-time `ibmq_jakarta` case.

Focusing on CPMG, for short times, it does slightly worse than the other sequences, yet still much better than Free, but for long times, it does about as poorly as Free. On `ibmq_bogota` and `ibmq_jakarta`, XY4 performs significantly worse than UR $_n$ , and QDD $_{n,m}$  and is always worse than

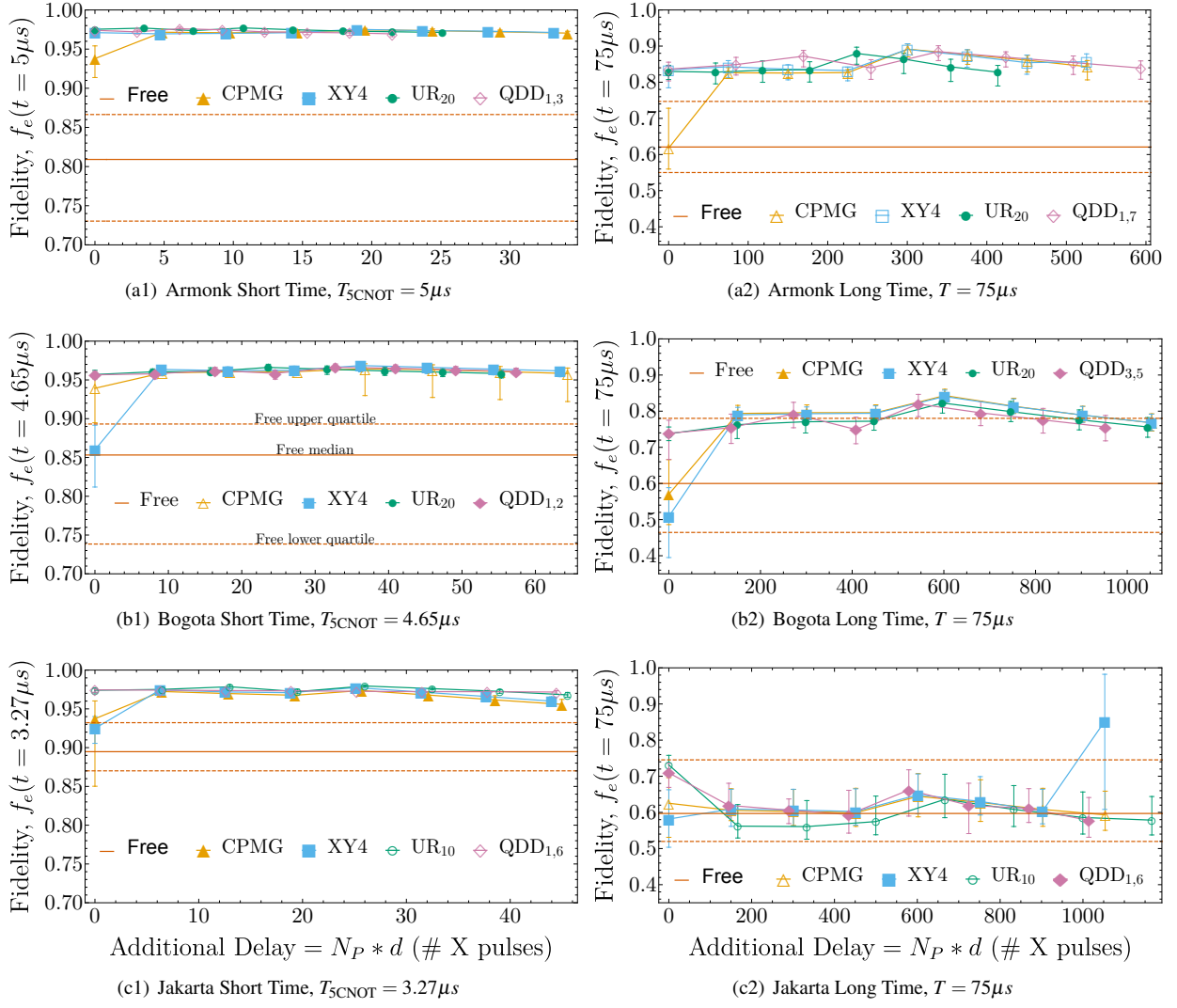


FIG. 4. Summary of the Haar interval experiment across the three devices. We explore the relationship between the median fidelity  $f_e(d, \zeta^*; T)$  across Haar random states as a function of pulse interval spacing,  $d$ . For a given device and  $T$ , we plot the best robust sequence ( $\text{UR}_n$ ) and the best non-uniform sequence  $\text{QDD}_{n,m}$  from Fig. 3, as well as CPMG, XY4, and Free evolution as consistent references.  $n, m$  in  $\text{QDD}_{n,m}$  is chosen so that the sequence fits the time window. The left and right column corresponds to short ( $T = T_{5\text{CNOT}}$ ) and long times ( $T = 75\mu\text{s}$ ) respectively. Confidence intervals are upper and lower quartiles – 75% and 25% of all fidelities lie below them, respectively. Both the asymmetric (open symbols) and symmetric (closed symbols) sequences were considered, but we display only the better of the two (for some empirically optimal  $d^*$ ).

CPMG. The main exception to this rule is `ibmq_armonk` where XY4 is comparable to  $\text{UR}_n$  and  $\text{QDD}_{n,m}$  in both the short and long time limits.

Overall, while all sequences lose effectiveness in the long-time limit, the advanced sequences perform well when the pulse spacing is  $d = 0$ .

## 2. $d > 0$ : Increasing the pulse interval can improve DD performance

It is clear from Fig. 4 (most notably from Fig. 4(b2)), that increasing the pulse interval can sometimes significantly im-

prove DD performance. For example, considering Fig. 4(a1), at  $d = 0$  CPMG is worse than the other sequences, but by increasing  $d$ , CPMG matches the performance of the sequences. The same qualitative result occurs even more dramatically for long times (Fig. 4(a2)). Here, CPMG goes from a median fidelity around 0.6 – as poor as Free – to around 0.9 at  $Npd = 300$ . The large improvement of CPMG with increasing  $d$  is fairly generic across devices and times, with the only exception being `ibmq_jakarta` and long times. Thus, even a simple sequence such as CPMG (or XY4, which behaves similarly) can compete with the highest ranking advanced sequences if the pulse-interval is optimized. It is unsurprising that optimizing pulse-interval can help, but the degree of im-

provement is surprising, particularly the ability of simple sequences to match the performance of advanced ones.

### 3. $d = d_{\max}$ : Performance at the single cycle limit

In a similar vein, it is notable how well sequences do in the  $d = d_{\max}$  limit (at the right of each plot in Fig. 4). For CPMG at  $T = 75\mu s$ , this corresponds to applying a pulse on average every  $37.5\mu s$ ; this certainly does not obey the principle of making  $\tau$  small, implied from error terms scaling as  $\mathcal{O}(\tau^n)$  as in the standard theory. There is always a tradeoff between combating errors and packing too many pulses on real devices with finite width and implementation errors. Moreover, the circuit structure will restrict the pulse interval when implementing DD on top of a quantum circuit. The general rule gleaned from Fig. 4 is to err toward the latter and apply DD sparsely. In particular,  $d = d_{\max}$  results in either a comparable performance to  $d = 0$  or a potentially large improvement.

Nevertheless, whether dense DD is better than sparse DD can depend on the specific device characteristics, desired timescale, and relevant metrics. As a case in point, note the Haar experiment results on `ibmq_jakarta` in the long-time limit. Here, most sequences – aside from XY4 – deteriorate with increasing  $d$ . Surprisingly, the best strategy in median performance is XY4, which does come at the cost of a large inner-quartile range.

$UR_n$  for  $d = 0$  does substantially better than Free for all six panels, which is an empirical confirmation of its claimed robustness. Even for  $d > 0$ ,  $UR_n$  remains a high fidelity and low variance sequence. Since other sequences only roughly match  $UR_n$  upon interval optimization, using a robust sequence is a safe default choice.

### C. Saturation of CDD and UDD, and an optimum for UR

In Fig. 5, we display the time-averaged fidelity from `ibmq_bogota` for  $CDD_n$ ,  $UR_n$  and  $UDD_{x_n}$  as a function of  $n$ . Related results for the other DD sequence families are discussed in Appendix D. As discussed in Section IA 2,  $CDD_n$  performance is expected to saturate at some  $n_{\text{opt}}$  according to Eq. (20). In Fig. 5(a), we observe evidence of this saturation at  $n_{\text{opt}} = 3$  on `ibmq_bogota`. We can use this to provide an estimate of  $\varepsilon = \|H_B\| + \|H_{SB}\|$ . Substituting  $n_{\text{opt}}$  into Eq. (20) we find:

$$\bar{\varepsilon}\varepsilon\Delta \in [4^{-5}, 4^{-4}] = [9.77 \times 10^{-4}, 3.91 \times 10^{-3}]. \quad (45)$$

This means that  $\varepsilon\Delta \ll 1$  (we set  $\bar{\varepsilon} \approx 1$ ), which confirms the assumption we needed to make for DD to give a reasonable suppression given that XY4 yields  $\mathcal{O}(\varepsilon\tau^2)$  suppression. This provides a level of empirical support for the validity of our assumptions. In addition,  $\Delta \approx 51$  ns on `ibmq_bogota`, so we conclude that  $\varepsilon \approx 0.5$  MHz. Since qubit frequencies are roughly  $\omega_q \approx 4.5 - 5$  GHz on IBM devices, this also confirms that  $\omega_q \gg \varepsilon$ , as required for a DD pulse. We observe a similar saturation in  $CDD_n$  on `Jakarta` and `Armonk` as well (Appendix D).

Likewise, for an ideal experiment with fixed time  $T$ , the performance of  $UDD_{x_n}$  should scale as  $\mathcal{O}(\tau^n)$ , and hence we expect a performance that increases monotonically  $n$ . In practice, this performance should saturate once the finite-pulse width error  $\mathcal{O}(\Delta)$  is the dominant noise contribution [59]. Once again, the  $UDD_n$  sequence performance on `ibmq_bogota` is consistent with theory. In particular, we expect (and observe) a consistent increase in performance with increasing  $n$  until performance saturates. While this saturation is also seen on `Armonk`, on `Jakarta`  $UDD_{x_n}$ 's performance differs significantly from theoretical expectations (see Appendix D).

$UR_n$  also experiences a tradeoff between error suppression and noise introduced by pulses. After a certain optimal  $n$ , the performance for  $UR_n$  is expected to drop [46]. In particular, while  $UR_n$  yields  $\mathcal{O}(\varepsilon_r^{n/2})$  suppression with respect to flip angle errors (Section IB 3), all  $UR_n$  provide  $\mathcal{O}(\tau^2)$  decoupling, i.e., adding more free evolution periods also means adding more noise. Thus, we expect  $UR_n$  to improve with increasing  $n$  until performance saturates. On `ibmq_bogota`, by increasing up to  $UR_{20}$ , we gain a large improvement over  $UR_{10}$ , but increasing further to  $UR_{50}$  or  $UR_{100}$  results in a small degradation in performance; see Fig. 5. A similar saturation occurs with `ibmq_jakarta` and `ibmq_armonk` (see Appendix D).

## IV. SUMMARY AND CONCLUSIONS

We performed an extensive survey of 10 DD families (a total of 60 sequences) across three superconducting IBM devices. In the first set of experiments (the Pauli experiment, Section II A), we tested how well these 60 sequences preserve the six Pauli eigenstates over relatively long times ( $25 - 75\mu s$ ). Motivated by theory, we used the smallest possible pulse interval for all sequences. We then chose the top-performing QDD and UR sequences from the Pauli experiment for each device, along with CPMG, XY4, and free evolution as baselines, and studied them extensively. In this second set of experiments (the Haar experiment Section II B), we considered 25 (fixed) Haar-random states for a wide range of pulse intervals,  $\tau$ .

In the Pauli experiment (Section II A), we ranked sequence performance by the median time-averaged fidelity at  $T = 75\mu s$ . This ranking is consistent with DD theory. The best performing sequence on each device substantially outperforms free evolution. Moreover, the expected deviation, quantified using the inner-quartile range of the average fidelity, was much smaller for DD than for free evolution. Finally, 29 out of the 30 best performing sequences were “advanced” DD sequences, explicitly designed to achieve high-order cancellation or robustness to control errors.

In the Haar-interval experiment (Section II B), we reported point-wise fidelity rather than the coarse-grained time-averaged fidelity. At  $\tau = 0$ , the Haar-interval experiment is identical to the Pauli experiment except for the expanded set of states. Indeed, we found the same hierarchy of sequence performance between the two experiments. For example, on `ibmq_jakarta`, we found  $XY4 < CPMG < QDD_{1,6} < UR_{10}$  for



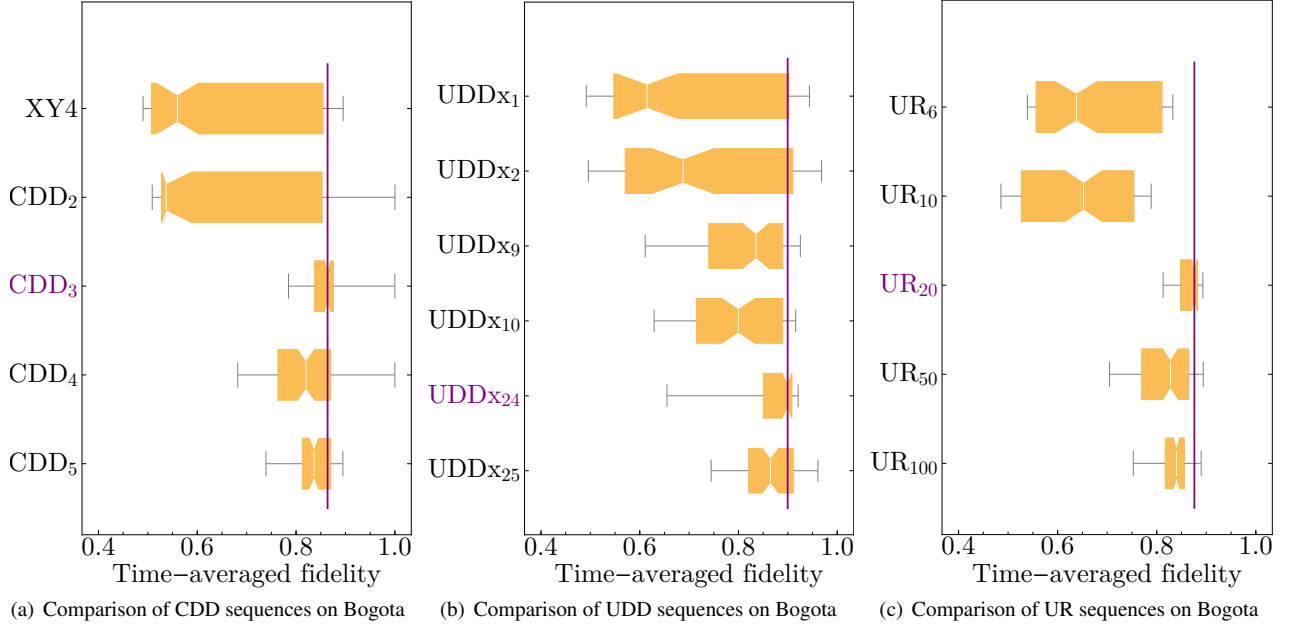


FIG. 5.  $f_e(t)$  Average fidelity at  $75\mu s$  for (a)  $CDD_n$  for  $n \in \{1, \dots, 5\}$ , (b)  $UDD_{x_n}$  for  $x \in \{1, 2, 9, 24, 25\}$ , (c)  $UR_n$  for  $n \in \{6, 10, 20, 50, 100\}$ . All three sequence families exhibit saturation on `ibmq_bogota` as we increase  $n$ , as expected from theory [34, 46, 59].

both Fig. 3 and Fig. 4. This suggests that a test over the Pauli states is a good proxy to Haar-random states for our metric.

However, once we allowed the pulse interval ( $\tau$ ) to vary, we found two unexpected results. First, contrary to expectations, advanced sequences, which theoretically provide a better performance, do not retain their performance edge. Second, for most devices and times probed, DD sequence performance improves or stays roughly constant with increasing pulse intervals before decreasing slightly for very long pulse intervals. This effect is particularly significant for the basic CPMG and XY4 sequences. Relating these two results, we found that with pulse-interval optimization, the basic sequences' performance is statistically indistinguishable from that of the advanced UR and QDD sequences. In stark contrast to the theoretical prediction favoring short intervals, choosing the longest possible pulse interval, with one sequence repetition within a given time, is generally better than the minimum interval. The one exception to these observations is the `ibmq_jakarta` processor for  $T = 75\mu s$ , which is larger than its mean  $T_2$  of  $20.7\mu s$ . Here, the advanced sequences significantly outperform the basic sequences at their respective optimal interval values ( $\tau = 0$  for the advanced sequences), and DD performance degrades with sufficiently large pulse intervals. The short  $T_2$  for `ibmq_jakarta` is notable, since in contrast,  $T = 75\mu s < \langle T_2 \rangle$  for both `ibmq_armonk` ( $\langle T_2 \rangle = 230\mu s$ ) and `ibmq_bogota` ( $\langle T_2 \rangle = 146\mu s$ ). We may this conclude that, overall, sparse DD is preferred to tightly-packed DD provided decoherence in the free evolution periods between pulses is not too strong.

The UR sequence either matched or nearly matched the best performance of any other tested sequence at any  $\tau$  for each

device. It also achieved near-optimal performance at  $\tau = 0$  in four out of the six cases shown in Fig. 4. This is a testament to its robustness and suggests the  $UR_n$  family is a good generic choice, provided an optimization to find a suitable  $n$  for a given device is performed. In our case, this meant choosing the top performing  $UR_n$  member from the Pauli experiment. Alternatively, our results suggest that as long as  $T < T_2$ , one can choose a basic sequence and likely achieve comparable performance by optimizing the pulse interval. In other words, *optimizing the pulse interval for a given basic DD sequence or optimizing the order of an advanced DD sequence at zero pulse interval are equally effective strategies for using DD in practice*. However, the preferred strategy depends on hardware constraints. For example, if OpenPulse access (or a comparable control option) is not available so that a faithful  $UR_n$  implementation is not possible, one would be constrained to optimizing CPMG or XY4 pulse intervals. Under such circumstances, where the number of DD optimization parameters is restricted, using a variational approach to identify the remaining parameters (e.g., pulse intervals) can be an effective approach [82].

Overall, theoretically promising, advanced DD sequences work well in practice. However, one must fine-tune the sequence to obtain the best DD performance for a given device. A natural and timely extension of our work would be developing a rigorous theoretical understanding of our observations, which do not always conform to previous theoretical expectations. Developing DD sequences for specific hardware derived using the physical model of the system instead of experimental optimization, or using machine learning methods [83], are other interesting directions. A thorough understanding of

how to tailor DD sequences to today's programmable quantum computers could be vital in using DD to accelerate the attainment of fault-tolerant quantum computing [34].

## ACKNOWLEDGMENTS

NE was supported by the U.S. Department of Energy (DOE) Computational Science Graduate Fellowship under Award Number DE-SC0020347. This material is based upon work supported by the National Science Foundation the Quantum Leap Big Idea under Grant No. OMA-1936388. LT and GQ acknowledge funding from the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Accelerated Research in Quantum Computing under Award Number DE-SC0020316. We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors, and do not reflect the official policy or position of IBM or the IBM Quantum team. We acknowledge the access to advanced services provided by the IBM Quantum Researchers Program. We are thankful to Vinay Tripathi for insightful discussions, particularly regarding device specific effects. D.L. is grateful to Prof. Lorenza Viola for insightful discussions and comments.

## Appendix A: Summary of the DD sequences benchmarked in this work

Here we provide definitions for all the DD sequences we tested. To clarify what free evolution periods belong between pulses, we treat uniform and non-uniform pulse interval sequences separately. When possible, we define a pulse sequence in terms of another sequence (or entire DD sequence) using the notation  $[\cdot]$ . In addition, several sequences are recursively built from simpler sequences. When this happens, we use the notation  $S = s_1([s_2])$ , whose meaning is illustrated by the example of CDD<sub>n</sub> [Eq. (15)].

### 1. Uniform pulse interval sequences

All the uniform pulse interval sequences are of the form

$$f_\tau - P_1 - f_{2\tau} - P_2 - f_{2\tau} - \dots - f_{2\tau} - P_n - f_\tau. \quad (\text{A1})$$

For brevity, we omit the the free evolution periods in the following definitions.

We distinguish between single and multi-axis sequences, by which we mean the number of orthogonal interactions in the system-bath interaction (e.g., pure dephasing is a single-axis case), not the number of axes used in the pulse sequences.

First, we list the single-axis DD sequences:

$$\text{Hahn} \equiv X \quad (\text{A2a})$$

$$\text{super-Hahn/RGA}_{2x} \equiv X - \bar{X} \quad (\text{A2b})$$

$$\text{RGA}_{2y} \equiv Y - \bar{Y} \quad (\text{A2c})$$

$$\text{CPMG} \equiv X - X \quad (\text{A2d})$$

$$\text{super-CPMG} \equiv X - X - \bar{X} - \bar{X}. \quad (\text{A2e})$$

Second, the UR<sub>n</sub> sequence for  $n \geq 4$  and  $n$  even is defined as

$$\text{UR}_n = (\pi)_{\phi_1} - (\pi)_{\phi_2} - \dots - (\pi)_{\phi_n} \quad (\text{A3a})$$

$$\phi_k = \frac{(k-1)(k-2)}{2} \Phi^{(n)} + (k-1)\phi_2 \quad (\text{A3b})$$

$$\Phi^{(4m)} = \frac{\pi}{m} \quad \Phi^{(4m+2)} = \frac{2m\pi}{2m+1}, \quad (\text{A3c})$$

where  $(\pi)_\phi$  is a  $\pi$  rotation about an axis which makes an angle  $\phi$  with the  $x$ -axis,  $\phi_1$  is a free parameter usually set to 0 by convention, and  $\phi_2 = \pi/2$  is a standard choice we use. This is done so that UR<sub>4</sub> = XY4 as discussed in Ref. [46] (note that despite this UR<sub>n</sub> was designed for single-axis decoherence).

Next, we list the multi-axis sequences. We start with XY4 and all its variations:

$$\text{XY4/CDD}_1 \equiv Y - X - Y - X \quad (\text{A4a})$$

$$\text{CDD}_n \equiv \text{XY4}([\text{CDD}_{n-1}]) \quad (\text{A4b})$$

$$\text{RGA}_4 \equiv \bar{Y} - X - \bar{Y} - X \quad (\text{A4c})$$

$$\text{RGA}_{4p} \equiv \bar{Y} - \bar{X} - \bar{Y} - \bar{X} \quad (\text{A4d})$$

$$\text{RGA}_{8c}/\text{XY8} \equiv X - Y - X - Y - Y - X - Y - X \quad (\text{A4e})$$

$$\text{RGA}_{8a} \equiv X - \bar{Y} - X - \bar{Y} - Y - \bar{X} - Y - \bar{X} \quad (\text{A4f})$$

$$\begin{aligned} \text{super-Euler} \equiv & X - Y - X - Y - Y - X - Y - X \\ & - \bar{X} - \bar{Y} - \bar{X} - \bar{Y} - \bar{Y} - \bar{X} - \bar{Y} - \bar{X} \end{aligned} \quad (\text{A4g})$$

$$\text{KDD} \equiv [K_{\pi/2}] - [K_0] - [K_{\pi/2}] - [K_0], \quad (\text{A4h})$$

where  $K_\phi$  is a composite of 5 pulses:

$$K_\phi \equiv (\pi)_{\pi/6+\phi} - (\pi)_\phi - (\pi)_{\pi/2+\phi} - (\pi)_\phi - (\pi)_{\pi/6+\phi}. \quad (\text{A5})$$

For example,  $(\pi)_0 = X$  and  $(\pi)_{\pi/2} = Y$ . The series  $K(\phi)$  itself is a  $\pi$  rotation about the  $\phi$  axis followed by a  $-\pi/3$  rotation about the  $z$ -axis. To see this, note that a  $\pi$  rotation about the  $\phi$  axis can be written  $(\pi)_\phi = R_Z(-\phi)R_Y(-\pi/2)R_Z(\pi)R_Y(\pi/2)R_Z(\phi)$ , and one can verify the claim by direct matrix multiplication. KDD (A4h) is the Knill-composite version of XY4 with a total of 20 pulses [40, 67]. Note that the alternation of  $\phi$  between 0 and  $\pi/2$  means that successive pairs give rise to a  $-\pi/3 + \pi/3 = 0$   $z$ -rotation at the end.

Next, we list the remaining multi-axis RGA sequences:

$$\text{RGA}_{16b} \equiv \text{RGA}_{4p}([\text{RGA}_{4p}]) \quad (\text{A6a})$$

$$\text{RGA}_{32a} \equiv \text{RGA}_4([\text{RGA}_{8a}]) \quad (\text{A6b})$$

$$\text{RGA}_{32c} \equiv \text{RGA}_{8c}([\text{RGA}_4]) \quad (\text{A6c})$$

$$\text{RGA}_{64a} \equiv \text{RGA}_{8a}([\text{RGA}_{8a}]) \quad (\text{A6d})$$

$$\text{RGA}_{64c} \equiv \text{RGA}_{8c}([\text{RGA}_{8c}]) \quad (\text{A6e})$$

$$\text{RGA}_{256a} \equiv \text{RGA}_4([\text{RGA}_{64a}]) \quad (\text{A6f})$$

## 2. Non-uniform pulse interval sequences

The non-uniform sequences are described by a general DD sequence of the form

$$f_{\tau_1} - P_1 - f_{\tau_2} - \dots - f_{\tau_m} - P_m - f_{\tau_{m+1}} \quad (\text{A7})$$

for pulses  $P_j$  applied at times  $t_j$  for  $1 \leq j \leq m$ . Thus for ideal, zero-width pulses, the interval times are  $\tau_j = t_j - t_{j-1}$  with  $t_0 \equiv 0$  and  $t_{m+1} = T$ , the total desired duration of the sequence.

### a. Ideal UDD

For ideal pulses, UDD $x_n$  is defined as follows. For a desired evolution time  $T$ , apply  $X$  pulses at times  $t_j$  given by

$$t_j = T \sin^2 \left( \frac{j\pi}{2n+2} \right), \quad (\text{A8})$$

for  $j = 1, 2, \dots, n$  if  $n$  is even and  $j = 1, 2, \dots, n+1$  if  $n$  is odd. Hence, UDD $x_n$  always uses an even number of pulses –  $n$  when  $n$  is even and  $n+1$  when  $n$  is odd – so that when no noise or errors are present UDD $x_n$  faithfully implements a net identity operator.

### b. Ideal QDD

For the purpose of defining QDD it is useful to instead define UDD $x_n$  in terms of the pulse intervals,  $\tau_j = t_j - t_{j-1}$ . By defining the normalized pulse interval,

$$s_j = \frac{t_j - t_{j-1}}{t_1} = \sin \left( \frac{(2j-1)\pi}{2n+2} \right) \csc \left( \frac{\pi}{2n+2} \right), \quad (\text{A9})$$

for  $j = 1, 2, \dots, n+1$ , we can define UDD $x_n$  over a total time  $T$ ,

$$\text{UDD}x_n(T) \equiv X^n - f_{s_{n+1}T} - X - f_{s_nT} - \dots - X - f_{s_1T}, \quad (\text{A10})$$

where the notation  $X^n$  means that the sequence ends with  $X$  ( $I$ ) for odd (even)  $n$ . From this, QDD $_{n,m}$  has the recursive definition

$$Y^n - \text{UDD}x_m(s_{n+1}T) - Y - \text{UDD}x_m(s_nT) - \dots - Y - \text{UDD}x_m(s_1T). \quad (\text{A11})$$

This means that we implement UDD $y_n$  (the outer  $Y$  pulses), and embed an  $m^{\text{th}}$  order UDD $x_m$  sequence within the free evolution periods of this sequence. The inner UDD $x_m$  sequences have a rescaled total evolution time  $s_kT$ , and since the decoupling properties only depend on  $\tau_j$  (and not the total time), we still obtain the expected inner cancellation. Written in this way, the total evolution time of QDD $_{n,m}$  is  $S_n^2T$  where  $S_n = \sum_{j=1}^{n+1} s_j$ .

To match the convention of all other sequences presented, we connect this definition to one in which the total evolution time of QDD $_{n,m}$  is itself  $T$ . First, we implement the outer

UDD $y_n$  sequence with  $Y$  pulses placed at times  $t_j$  according to Eq. (A8). The inner  $X$  pulses must now be applied at times

$$t_{j,k} = \tau_j \sin^2 \left( \frac{k\pi}{2n+2} \right) + t_{j-1}, \quad (\text{A12})$$

where  $j = 1, 2, \dots, n$  if  $n$  is even (or  $j = 1, 2, \dots, n+1$  if  $n$  is odd), with a similar condition for  $k$  up to  $m$  if  $m$  is even (or  $m+1$  if  $m$  is odd). Note that when  $m$  is odd, we end each inner sequence with an  $X$  and then the outer sequence starts where a  $Y$  must be placed at the same time. In these cases, we must apply a  $Z = XY$  pulse (ignoring the global phase which does not affect DD performance). Hence, we must apply rotations about  $X, Y$ , and  $Z$  when  $m$  is odd.

### c. UDD and QDD with finite-width pulses

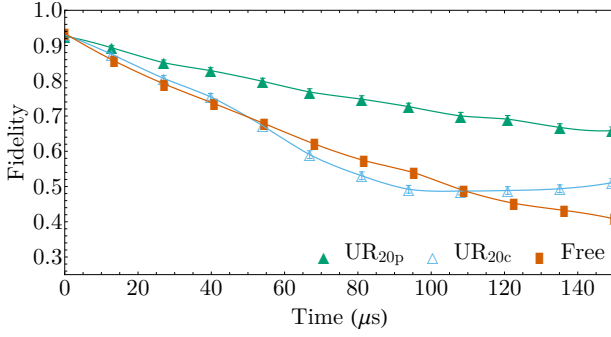
For real pulses with finite width  $\Delta$ , these formulas must be slightly augmented. First, defining  $t_j$  is ambiguous since the pulse application cannot be instantaneous at time  $t_j$ . In our implementation, pulses start at time  $t = t_j - \Delta$  so that they end at the time when they should be applied in the ideal case. Trying two other strategies – placing the center of the pulse at  $t_j$  or the beginning of the pulse at  $t_j$  – did not result in a noticeable difference. Furthermore,  $X$  and  $Y$  have finite width (roughly  $\Delta = 50$  ns). When UDD $x_n$  is applied for  $n$  even, we must end on an identity, so the identity must last for a duration  $\Delta$ , i.e.,  $I = f_{\tau}$ . A similar timing constraint detail appears for QDD $_{n,m}$  when  $m$  is odd. Here, we must apply a  $Z$  pulse, but on IBM devices,  $Z$  is virtual and instantaneous (see Appendix B). Thus, we apply  $Z - f_{\Delta}$  to obtain the expected timings.

## Appendix B: Circuit vs OpenPulse APIs

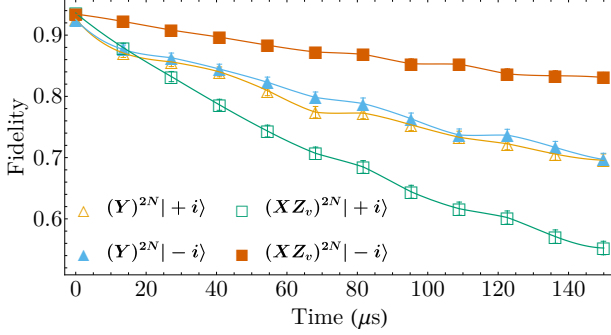
We first tried to use the standard Qiskit circuit API [69, 70]. Given a DD sequence, we transpiled the circuit Fig. 1 to the respective device's native gates. However, as we illustrate in Fig. 6(a), this can lead to many advanced sequences, such as UR<sub>20</sub>, behaving worse than expected. Specifically, this figure shows that implementing UR<sub>20</sub> in the standard circuit way, denoted UR<sub>20c</sub> (where c stands for circuit), is substantially worse than an alternative denoted UR<sub>20p</sub> (where p stands for pulse).

The better alternative is to use OpenPulse [68]. We call this the “pulse” implementation of a DD sequence. The programming specifics are provided in Ref. [84]; here, we focus on the practical difference between the two methods with the illustrative example shown in Fig. 6(b). Specifically, we compare the  $Y f_{\tau} Y f_{\tau}$  sequence implemented in the circuit API and the OpenPulse API.

Under OpenPulse, the decay profiles for  $|+i\rangle$  and  $|-i\rangle$  are roughly identical, as expected for the  $Y f_{\tau} Y f_{\tau}$  sequence. The slight discrepancy can be understood as arising from coherent errors in state preparation and the subsequent  $Y$  pulses, which accumulate over many repetitions. On the other hand, the circuit results exhibit a large asymmetry between  $|i\rangle$  and



(a) Comparison of pulse and circuit implementation of UR<sub>20</sub> on ibmq\_armonk with respect to Free



(b) Comparison of pulse and circuit implementation of  $Y - Y$  on ibmq\_armonk

FIG. 6. Comparison of the circuit and OpenPulse API approaches to implementing (a) UR<sub>20</sub> and (b)  $Y f_\tau Y f_\tau$ . Panel (a) demonstrates that the OpenPulse version is substantially better. This is partially explained by (b). When using OpenPulse, the  $Y f_\tau Y f_\tau$  sequence behaves as expected by symmetrically protecting  $|\pm i\rangle$ , in stark contrast to the circuit implementation, which uses virtual Z gates, denoted  $Z_v$ .

$|-i\rangle$ . The reason is that  $Y$  is compiled into  $Z_v - X$  where  $Z_v$  denotes a virtual Z gate [85]. As Fig. 6(b) shows,  $Z_v - X$  does *not* behave like  $Y$ . The simplest explanation consistent with the results is to interpret  $Z_v$  as an instantaneous  $Z$ . In this case,  $Z|+i\rangle = |-i\rangle$  and the subsequent  $X$  rotates the state from  $|-i\rangle$  to  $|+i\rangle$  by rotating *through the excited state*. The initial state  $|-i\rangle$ , on the other hand, rotates *through the ground state*. Since the ground state is much more stable than the excited state on IBMQ's transmon devices, this asymmetry in trajectory on the Bloch sphere is sufficient to explain the asymmetry in fidelity.<sup>8</sup>

Taking a step back, every gate that is not a simple rotation about the  $x$  axis is compiled by the standard circuit approach into one that is a combination of  $Z_v$ ,  $X$ , and  $\sqrt{X}$ . These gates can behave unexpectedly, as shown here. In addition, the transpiler – unless explicitly instructed otherwise – also sometimes combines a  $Z_v$  into a global phase without implementing it right before an  $X$  gate. Consequently, two circuits can be logically equivalent while implementing different DD

sequences. Using OpenPulse, we can ensure the proper implementation of  $(\pi)_\phi$ . This allows the fidelity of UR<sub>20p</sub> to exceed that of UR<sub>20c</sub>.

Overall, we found (not shown) that the OpenPulse implementation was almost always better than or comparable to the equivalent circuit implementation, except for XY4 on ibmq\_armonk, where XY4<sub>c</sub> was substantially better than XY4<sub>p</sub>. However, XY4<sub>p</sub> was not the top performing sequence. Hence, it seems reasonable to default to using OpenPulse for DD when available.

### Appendix C: Toggling frame

We assume that without any external control, the system and bath evolve under the time-independent noise Hamiltonian  $H$ . A DD pulse sequence is realized via a time-dependent control Hamiltonian  $H_c(t)$  acting only on the system so that the system and bath evolve according to  $H + H_c(t)$  (refer to Section I A for definitions of the various Hamiltonian terms).

For understanding the effects of the control Hamiltonian, it is convenient to use the interaction picture defined by  $H_c(t)$ , also known as the *toggling frame* [16, 26, 29, 86, 87]. The toggling-frame density operator  $\tilde{\rho}_{SB}(t)$  is related to the Schrödinger-picture density operator  $\rho_{SB}(t)$  by

$$\begin{aligned} \rho_{SB}(t) &= U(t, 0) \rho_{SB}(0) U^\dagger(t, 0) \\ &\equiv U_c(t) \tilde{\rho}_{SB}(t) U_c^\dagger(t), \end{aligned} \quad (C1)$$

where  $U(t, 0)$  is the evolution operator generated by the full Hamiltonian  $H + H_c(t)$ . Therefore the toggling-frame state evolves according to

$$\tilde{\rho}_{SB}(t) = \tilde{U}(t, 0) \tilde{\rho}_{SB}(0) \tilde{U}^\dagger(t, 0), \quad (C2)$$

where the toggling-frame time evolution operator

$$\tilde{U}(t, 0) \equiv U_c^\dagger(t) U(t, 0) \quad (C3)$$

is generated by the *toggling-frame Hamiltonian*

$$\tilde{H}(t) \equiv U_c^\dagger(t) H U_c(t). \quad (C4)$$

Since  $U_c(t)$  acts nontrivially only on the system,  $\tilde{H}(t)$  can be written as

$$\tilde{H}(t) = H_B + \tilde{H}_{\text{err}}(t), \quad (C5)$$

where  $\tilde{H}_{\text{err}}(t) \equiv U_c^\dagger(t) H_{\text{err}} U_c(t)$  is the toggling-frame version of  $H_{\text{err}}$ . Because the operator norm is unitarily-invariant, we have  $\|\tilde{H}(t)\| = \|H\| \leq \epsilon$  and  $\|\tilde{H}_{\text{err}}(t)\| = \|H_{\text{err}}\| \leq J$ .

Throughout, we consider *cyclic DD*, where  $U_c(t)$  returns to the identity (up to a possible irrelevant overall phase) at the end of a cycle taking time  $t_{\text{DD}}$ :

$$U_c(t_{\text{DD}}) = U_c(0) = I. \quad (C6)$$

Therefore, at the end of the cycle, the toggling-frame and Schrödinger-picture states coincide.

<sup>8</sup> This observation and explanation is due to Vinay Tripathi.



## Appendix D: Results of all Pauli experiments

In Fig. 3, we summarized the results of the Pauli experiment for the top 10 sequences on each device and the baseline of CPMG, XY4, and free evolution. This appendix presents the data for all 60 tested sequences. In particular, we split the data for each device into a  $3 \times 3$  grid of plots shown in Figs. 7-9, separating by family as in Table I when possible. For convenience, CPMG, XY4, and Free are still included in all plots along with colored reference lines matching the convention in Fig. 3. The purple reference line denotes the best sequence in the given family plot excluding the baseline sequences placed at the end. For example, in Fig. 7(b)  $\text{RGA}_{64a}$  is the best RGA sequence even though it does not outperform XY4; it is marked with a purple reference line.

We plot the same families for each device in a  $3 \times 3$  grid, labeled (a)-(i). In each caption, we make a few general comments on DD performance overall and then comment on observations specific to each sequence family. Recall that a specific definition of each sequence is given in Appendix A, and a summary of their properties along with references in Table I. To avoid excessive repetition in each caption, we first provide a brief description of each of the cases (a)-(i) shown in Fig. 7-9:

- (a) This “family” serves as a catch-all for the basic sequences Hahn [47], CPMG [49, 64], and XY4 [48], along with sequences born from their modifications. Namely, we also plot the Eulerian super-cycle versions [65, 72] (denoted by S), KDD [40, 67] which is a composite pulse XY4, and  $\text{CDD}_n$  [42, 80] which is a recursive embedding of XY4.
- (b) The robust genetic algorithm (RGA) family [66].
- (c) The universally robust (UR) family [46].
- (d) The Uhrig dynamical decoupling (UDD) family using  $X$  pulses,  $\text{UDD}_x$  [43].
- (e-i) The quadratic dynamical decoupling (QDD) family with same inner and outer order,  $\text{QDD}_{n,n}$ , with fixed outer order 1,  $\text{QDD}_{1,m}$  with fixed outer order 2,  $\text{QDD}_{2,m}$  with fixed outer order 3,  $\text{QDD}_{3,m}$ , and with fixed outer order 4,  $\text{QDD}_{4,m}$ , respectively [44].

For each family, we expect the empirical hierarchy of sequence performance to be a complicated function of device-specific properties. Specifically, actual performance is a competition between (i) error cancellation order, (ii) number of free evolution periods, and (iii) systematic pulse errors due to finite width and miscalibrations, among other factors discussed in Section I. For each family, we summarize our expectations regarding these factors

- (a) For ideal pulses, we expect  $\text{CDD}_{n+1} > \text{CDD}_n \geq \text{KDD} = \text{S-XY4} = \text{XY4} > \text{S-CPMG} = \text{CPMG} = \text{S-Hahn} > \text{Hahn}$ . With a finite bandwidth constraint, we expect  $\text{CDD}_{n+1} > \text{CDD}_n$  to only hold up until some optimal concatenation level  $n_{\text{opt}}$  after which performance

saturates. Using finite width pulses with systematic errors, we expect  $\text{S-Hahn} > \text{Hahn}$  (and similarly for other S sequences) and  $\text{KDD} > \text{XY4}$  provided the additional robustness is helpful. I.e., if the pulses are extremely well calibrated and errors are dominated by latent bath-induced errors, then we should instead see  $\text{Hahn} > \text{S-Hahn}$ .

- (b) The expected performance hierarchy for RGA is rather complicated, as indicated by the labeling, and is best summarized in depth using Table II of Ref. [66]. A quick summary is that if we have strong pulses dominated by miscalibration errors ( $\epsilon\Delta \ll \epsilon_r$ ), then we expect  $\text{RGA}_{8a}$  and  $\text{RGA}_{64a}$  to do well. In the opposite limit, we expect  $\text{RGA}_4, \text{RGA}_{8c}, \text{RGA}_{16b}, \text{RGA}_{64c}$  to do well. The increasing number indicates the number of pulses, and as this increases, the decoupling order  $\mathcal{O}(\tau^n)$  increases, and the same competition between order-cancellation and free evolution periods as in  $\text{CDD}_n$  also applies.
- (c) The  $\text{UR}_n$  sequence provides  $\mathcal{O}(\epsilon_r^{n/2})$  suppression of flip angle errors at the expense of using  $n$  free evolution periods. The relationship of  $n$  to  $\mathcal{O}(\tau)$  decoupling is not well established in Ref. [46], but by construction seems to be  $\mathcal{O}(\tau^2)$  for all  $n$ . Thus our expectation is that  $\text{UR}_n$  improves with increasing  $n$  until performance saturates and the  $\mathcal{O}(\tau^2)$  contribution dominates the  $\mathcal{O}(\epsilon_r^{n/2})$  contribution. To see this, note that for a fixed time, the number of free evolution periods will be roughly the same regardless of  $n$ .
- (d) Ideally, for a fixed experiment duration  $T$ , the performance of  $\text{UDD}_x$  should scale as  $\mathcal{O}(\tau^n)$ , and hence improves monotonically with increasing  $n$ . In practice, this performance should saturate once the finite-pulse width error  $\mathcal{O}(\Delta)$  is the dominant noise contribution.
- (e-i) An extensive numerical study of  $\text{QDD}_{n,m}$  performance is discussed in Ref. [58] with corresponding rigorous proofs in Ref. [60]. For ideal pulses, the decoupling order is expected to be at least  $\mathcal{O}(t_s^{\min\{m,n\}})$  where  $t_s$  is the total evolution time of implementing a single repetition. Since we are instead interested in a fixed total time  $T$  consisting of multiple sequence repetitions with a minimal pulse interval, the interplay of competing factors is quite complicated. Further, we are forced to apply rotations about  $X$ ,  $Y$ , and  $Z$  to implement  $\text{QDD}_{n,m}$  when  $m$  is odd, but as noted in Appendix B,  $Z$  is virtual without OpenPulse. In summary, the naive theoretical expectation is that  $\text{QDD}_{n,m}$  should improve with increasing  $\min\{n,m\}$ , eventually saturating for the same reasons as  $\text{UDD}_x$ . However, we expect the fixed  $T$  and virtual- $Z$  set-up to complicate the actual results.

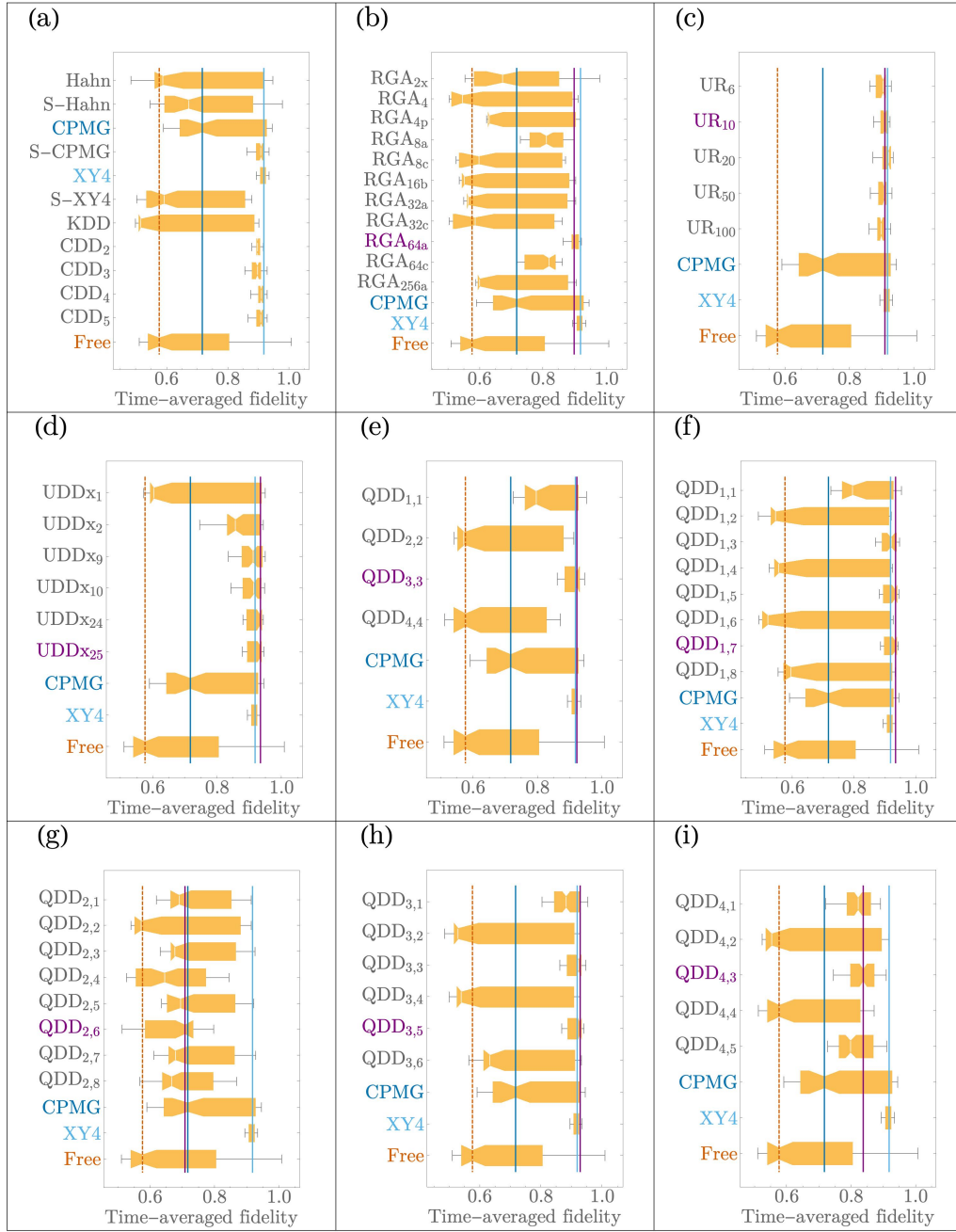


FIG. 7. Collection of all Pauli-experiment results for ibmq\_armonk. The best performing sequence for each family (solid purple line) substantially outperforms Free (orange) and CPMG (blue) but only marginally differs from the performance of XY4 (cyan).

(a) Results are not consistent with ideal-pulse theory but are sensible when considering realistic competing factors. First, S-Hahn  $>$  Hahn and S-CPMG  $>$  CPMG are consistent with the large pulse widths on ibmq\_armonk, for which  $\Delta \approx 142$  ns. That XY4 works very well is also consistent with its expected approximate universality given that  $T_1 \approx \frac{1}{2}T_2$  on ibmq\_armonk. Given that XY4 already performs well, it is not surprising that its robust versions, S-XY4 and KDD, do worse. In particular, they have little room for improvement but also add extra free evolution periods that accumulate additional error. Finally, CDD $_n$  is roughly flat for all  $n$  tested, which is consistent with an expected saturation that happens to occur at  $n_{\text{opt}} = 1$ .

(b) The performance of RGA does not match theoretical expectations. To see this, note that RGA $_4$  is itself a four-pulse sequence with error scaling  $\mathcal{O}(\tau^2)$  which is the same as XY4. Hence, it should perform comparably to XY4, but it does significantly worse. Furthermore, it is also unexpected that the best RGA sequences are 8a, 64a, and 64c. Indeed, 8a and 64a are expected to work best in a flip-angle error dominated regime, but in this regime, 64c has a scaling of  $\mathcal{O}(\epsilon_r^2)$ , so it is not expected to do well.

(c) The UR $_n$  sequence performance is consistent with theory. First note that UR $_4 = \text{XY4}$ . Thus, we expect UR $_n$  for  $n > 4$  to improve upon or saturate at the performance of XY4, and the latter is what happens.

(d) The UDD $_n$  sequence performance is consistent with theory. In particular, we expect (and observe) a consistent increase in performance with increasing  $n$  until performance saturates.

(e-i) The QDD $_{n,m}$  results in part match theoretical expectations, since they exhibit a strong even-odd effect, as predicted in Refs. [58, 60, 63]. Nevertheless, the optimal choice of  $n$  and  $m$  has to be fine-tuned for ibmq\_armonk. We note that five out of ten of the top ten sequences on ibmq\_armonk are from the QDD family.

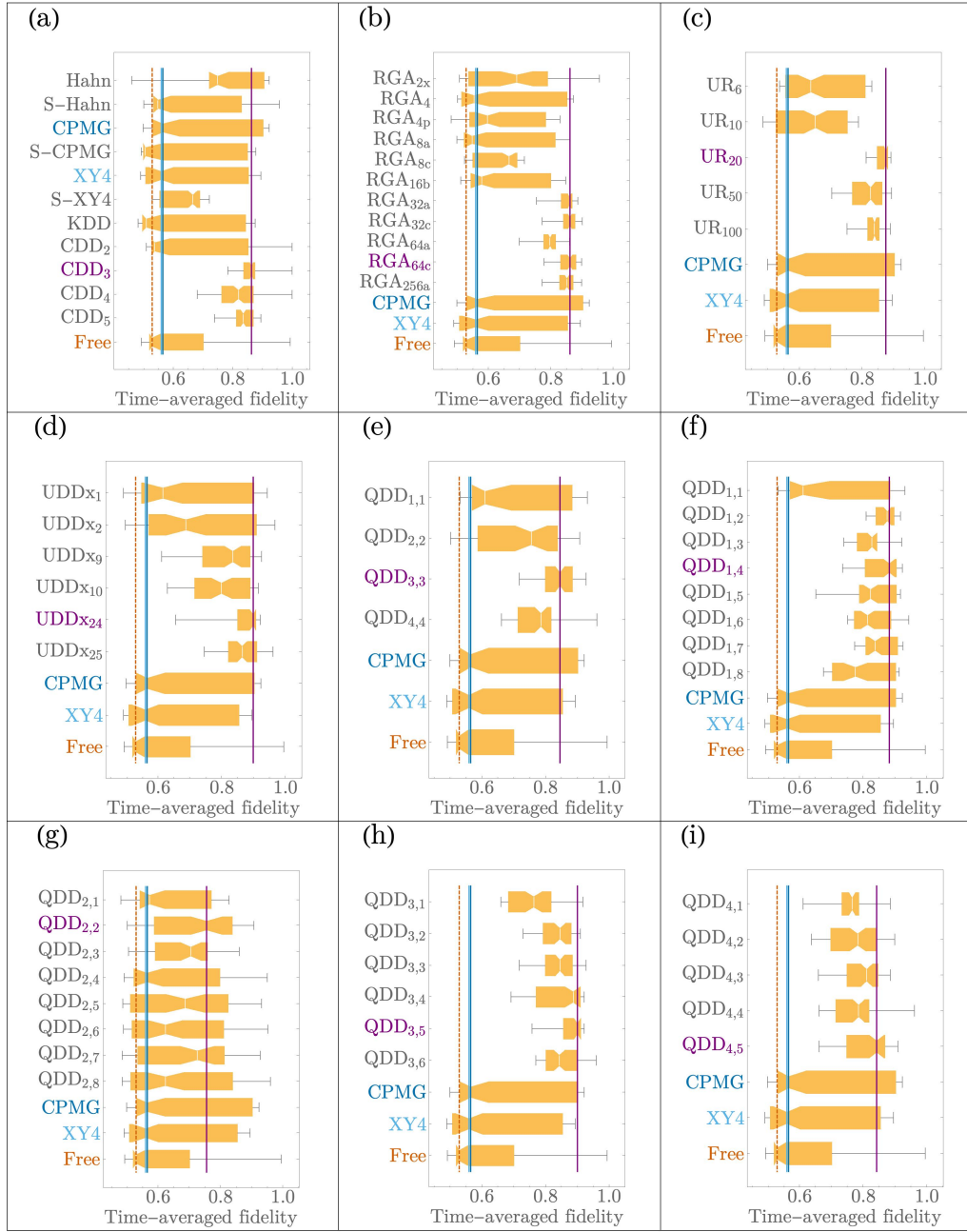


FIG. 8. Collection of all Pauli-experiment results for ibmq\_bogota. The best performing sequence for each family (solid purple line) substantially outperforms Free (orange), CPMG (blue), and XY4 (cyan).

(a) Results are not totally consistent with ideal-pulse theory. For example, it is unexpected that Hahn > CPMG and that Hahn < S-Hahn while at the same time S-XY4 > XY4. Nevertheless, some trends are expected such as CDD<sub>1</sub> < CDD<sub>2</sub> < CDD<sub>3</sub> and then saturating.

(b) Results are somewhat consistent with ideal-pulse theory. First of all, RGA<sub>4</sub> ≈ XY4 in performance, which is sensible. The first large improvement comes from RGA<sub>8c</sub> and then from numbers 32 and greater. This trend is similar to CDD<sub>n</sub> increasing, which is expected since larger RGA sequences are also recursively defined (e.g., RGA<sub>64c</sub> is a recursive embedding of RGA<sub>8c</sub> into itself). However, it is again unexpected that both ‘a’ and ‘c’ sequences should work well at the same time. For example, RGA<sub>8c</sub> > RGA<sub>8a</sub> theoretically means that ibmq\_bogota has negligible flip angle error. In such a regime, the decoupling order of RGA<sub>8a</sub> is the same as RGA<sub>64a</sub> since they are designed to cancel flip-angle errors. But, we find that RGA<sub>64a</sub> > RGA<sub>8a</sub> in practice.

(c) The UR<sub>n</sub> results are mostly consistent with theory. First, UR<sub>6</sub> is an improvement over XY4, and though UR<sub>n</sub> does increase with larger n, it is not simply monotonic as one would expect in theory. Instead, we find a more general trend with an empirical optimum at n = 20.

(d) The UDD<sub>xn</sub> results are mostly consistent with expectations. Again, performance mostly increases with increasing n, but the increase is not fully monotonic.

(e-i) The QDD<sub>n,m</sub> results are fairly consistent with theory. In (e), performance of QDD<sub>n,n</sub> increases with n until n = 3. The degradation for n = 4 is consistent with expectations in the bandwidth-limited setting [61]. In (f - i) the results are again fairly expected: aside from parity effects (odd/even m), for QDD<sub>n,m</sub>, we expect monotonic improvement with increasing m until n = m, after which performance should saturate or even slightly improve; this is the general empirical trend.

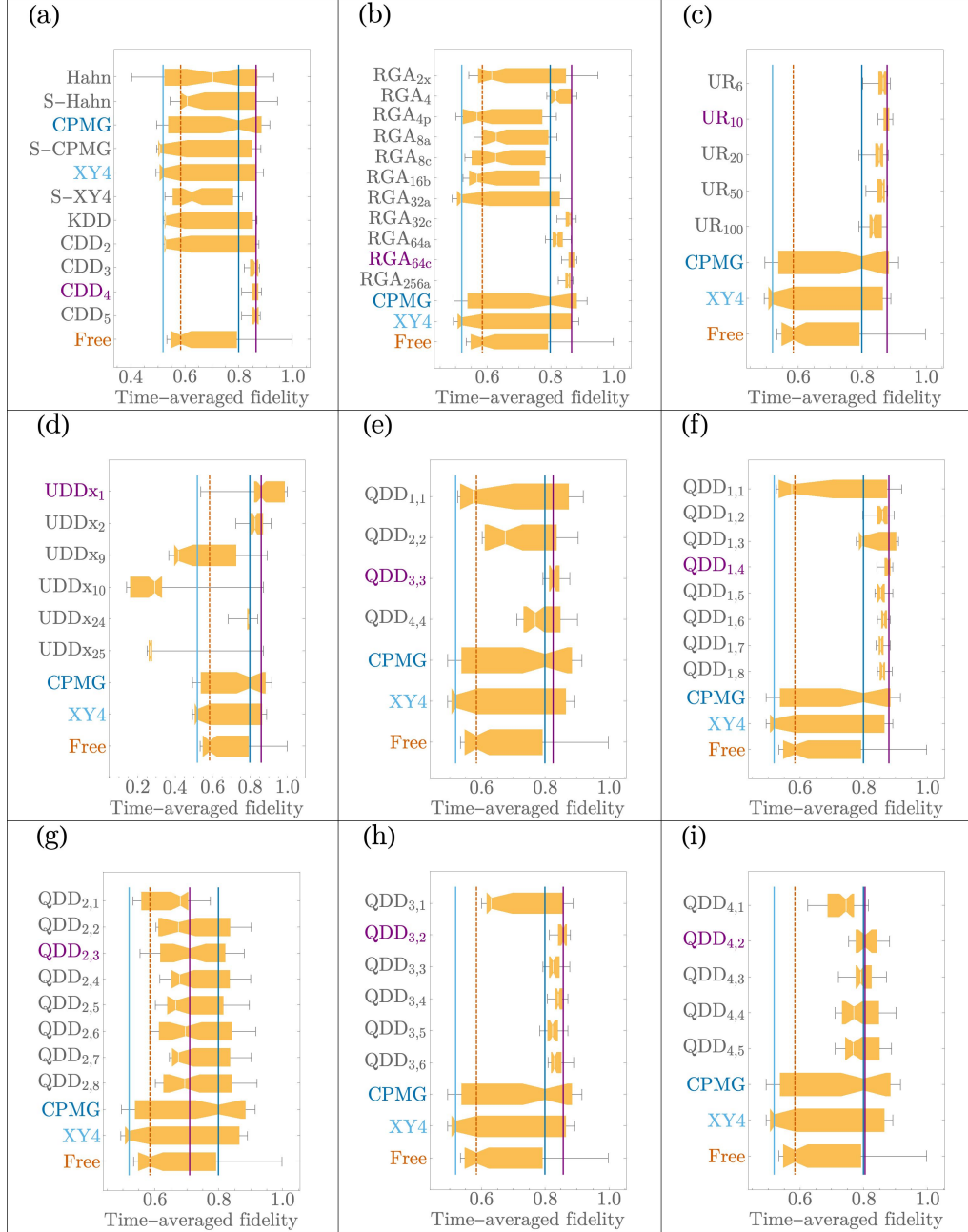


FIG. 9. Collection of all Pauli-experiment results for ibmq\_jakarta. The best performing sequence for each family (solid purple line) substantially outperforms Free (orange) and CPMG (blue) but only marginally differs from the performance of XY4 (cyan).

(a) Results are not fully consistent with ideal-pulse theory. For example, it is unexpected that Hahn  $>$  CPMG and that Hahn  $>$  S-Hahn while at the same time S-XY4  $>$  XY4. Nevertheless, some trends are expected, such as  $CDD_1 < CDD_2 < CDD_3 > CDD_4$  and then saturating.

(b) Results are somewhat consistent with the theory. First of all,  $RGA_4 > XY4$  is sensible and implies that we are in a flip-angle error dominated regime. However, we would then expect  $RGA_4 > RGA_{8a}$ , which does not occur. Nevertheless, the recursively defined sequences (number 32 and above) generally outperform their shorter counterparts, which is similar to  $CDD_n$  as expected.

(c) The  $UR_n$  results are consistent with theory. First,  $UR_6$  is a large improvement over XY4, and from there  $UR_{10} > UR_6$ . After this, the improvement plateaus.

(d) The performance of  $UDDx_n$  greatly differs from the theoretical expectation of monotonic improvement with  $n$ . In fact, the behavior is so erratic that we suspect device calibration errors dominated the experiment. Nevertheless, the performance of  $UDDx_1$  was excellent in this case.

(e-i) The  $QDD_{n,m}$  results are fairly consistent with theory, and quite similar to Fig. 8. The same comments as made there apply here.



- 
- [1] K. L. Pudenz, T. Albash, and D. A. Lidar, *Nat. Commun.* **5**, 3243 (2014).
- [2] S. J. Devitt, *Physical Review A* **94**, 032329 (2016).
- [3] W. Vinci, T. Albash, and D. A. Lidar, *npj Quant. Inf.* **2**, 16017 (2016).
- [4] J. R. Wootton and D. Loss, *Physical Review A* **97**, 052313 (2018).
- [5] C. Vuillot, *Quantum Inf. Comput.* **18**, 0949 (2018).
- [6] J. Roffe, D. Headley, N. Chancellor, D. Horsman, and V. Kendon, *Quantum Science and Technology* **3**, 035010 (2018).
- [7] D. Willsch, M. Willsch, F. Jin, H. De Raedt, and K. Michielsen, *Physical Review A* **98**, 052348 (2018).
- [8] A. Pearson, A. Mishra, I. Hen, and D. A. Lidar, *npj Quantum Information* **5**, 107 (2019).
- [9] R. Harper and S. T. Flammia, *Physical Review Letters* **122**, 080504 (2019).
- [10] C. Ryan-Anderson, J. Bohnet, K. Lee, D. Gresh, A. Hankin, J. Gaebler, D. Francois, A. Chernoguzov, D. Lucchetti, N. Brown, *et al.*, *Physical Review X* **11**, 041058 (2021).
- [11] Z. Chen, K. J. Satzinger, J. Atalaya, A. N. Korotkov, A. Dunsforth, D. Sank, C. Quintana, M. McEwen, R. Barends, P. V. Klimov, S. Hong, C. Jones, A. Petukhov, D. Kafri, S. Demura, B. Burkett, C. Gidney, A. G. Fowler, A. Paler, H. Putterman, I. Aleiner, F. Arute, K. Arya, R. Babbush, J. C. Bardin, A. Bengtsson, A. Bourassa, M. Broughton, B. B. Buckley, D. A. Buell, N. Bushnell, B. Chiaro, R. Collins, W. Courtney, A. R. Derk, D. Eppens, C. Erickson, E. Farhi, B. Foxen, M. Giustina, A. Greene, J. A. Gross, M. P. Harrigan, S. D. Harrington, J. Hilton, A. Ho, T. Huang, W. J. Huggins, L. B. Ioffe, S. V. Isakov, E. Jeffrey, Z. Jiang, K. Kechedzhi, S. Kim, A. Kitaev, F. Kostritsa, D. Landhuis, P. Laptev, E. Lucero, O. Martin, J. R. McClean, T. McCourt, X. Mi, K. C. Miao, M. Mohseni, S. Montazeri, W. Mruczkiewicz, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Newman, M. Y. Niu, T. E. O'Brien, A. Opremcak, E. Ostby, B. Pató, N. Redd, P. Roushan, N. C. Rubin, V. Shvarts, D. Strain, M. Szalay, M. D. Trevithick, B. Villalonga, T. White, Z. J. Yao, P. Yeh, J. Yoo, A. Zalcman, H. Neven, S. Boixo, V. Smelyanskiy, Y. Chen, A. Megrant, J. Kelly, and G. Q. AI, *Nature* **595**, 383 (2021).
- [12] L. Viola and S. Lloyd, *Phys. Rev. A* **58**, 2733 (1998).
- [13] P. Zanardi, *Physics Letters A* **258**, 77 (1999).
- [14] D. Vitali and P. Tombesi, *Physical Review A* **59**, 4178 (1999).
- [15] L.-M. Duan and G.-C. Guo, *Physics Letters A* **261**, 139 (1999).
- [16] L. Viola, E. Knill, and S. Lloyd, *Physical Review Letters* **82**, 2417 (1999).
- [17] M. J. Biercuk, H. Uys, A. P. VanDevender, N. Shiga, W. M. Itano, and J. J. Bollinger, *Nature* **458**, 996 (2009).
- [18] M. J. Biercuk, H. Uys, A. P. VanDevender, N. Shiga, W. M. Itano, and J. J. Bollinger, *Phys. Rev. A* **79**, 062324 (2009).
- [19] S. Damodarakurup, M. Lucamarini, G. D. Giuseppe, D. Vitali, and P. Tombesi, *Phys. Rev. Lett.* **103**, 040502 (2009).
- [20] J. Du, X. Rong, N. Zhao, Y. Wang, J. Yang, and R. B. Liu, *Nature* **461**, 1265 (2009).
- [21] E. R. Jenista, A. M. Stokes, R. T. Branca, and W. S. Warren, *J. Chem. Phys.* **131**, 204510 (2009).
- [22] G. A. Álvarez, A. Ajoy, X. Peng, and D. Suter, *Phys. Rev. A* **82**, 042306 (2010).
- [23] X. Peng, D. Suter, and D. A. Lidar, *Journal of Physics B: Atomic, Molecular and Optical Physics* **44**, 154003 (2011).
- [24] Y. Sagi, I. Almog, and N. Davidson, in *CLEO/QELS: 2010 Laser Science to Photonic Applications* (2010) pp. 1–2.
- [25] Z.-H. Wang, G. de Lange, D. Ristè, R. Hanson, and V. V. Dobrovitski, *Physical Review B* **85**, 155204 (2012).
- [26] L. Viola, S. Lloyd, and E. Knill, *Phys. Rev. Lett.* **83**, 4888 (1999).
- [27] L. Viola, *Phys. Rev. A* **66**, 012307 (2002).
- [28] K. Khodjasteh and D. A. Lidar, *Physical Review Letters* **89**, 197904 (2002).
- [29] L. Viola and E. Knill, *Phys. Rev. Lett.* **94**, 060502 (2005).
- [30] K. Khodjasteh and D. A. Lidar, *Physical Review A* **78**, 012355 (2008).
- [31] D. A. Lidar, *Phys. Rev. Lett.* **100**, 160506 (2008).
- [32] J. R. West, D. A. Lidar, B. H. Fong, and M. F. Gyure, *Phys. Rev. Lett.* **105**, 230503 (2010).
- [33] T. van der Sar, Z. H. Wang, M. S. Blok, H. Bernien, T. H. Taminiau, D. M. Toyli, D. A. Lidar, D. D. Awschalom, R. Hanson, and V. V. Dobrovitski, *Nature* **484**, 82 (2012).
- [34] H. K. Ng, D. A. Lidar, and J. Preskill, *Phys. Rev. A* **84**, 012305 (2011).
- [35] G. A. Paz-Silva and D. A. Lidar, *Sci. Rep.* **3** (2013).
- [36] B. Pokharel, N. Anand, B. Fortman, and D. A. Lidar, *Physical Review Letters* **121**, 220502 (2018).
- [37] A. M. Souza, *Process tomography of robust dynamical decoupling in superconducting qubits* (2020), arXiv:2006.10585 [quant-ph].
- [38] P. Jurcevic, A. Javadi-Abhari, L. S. Bishop, I. Lauer, D. F. Bogorin, M. Brink, L. Capelluto, O. Günlük, T. Itoko, N. Kanazawa, A. Kandala, G. A. Keefe, K. Krsulich, W. Landers, E. P. Lewandowski, D. T. McClure, G. Nannicini, A. Narasgond, H. M. Nayfeh, E. Pritchett, M. B. Rothwell, S. Srinivasan, N. Sundaresan, C. Wang, K. X. Wei, C. J. Wood, J.-B. Yau, E. J. Zhang, O. E. Dial, J. M. Chow, and J. M. Gambetta, *Quantum Sci. Technol.* **6**, 025020 (2021).
- [39] V. Tripathi, H. Chen, M. Khezri, K.-W. Yip, E. M. Levenson-Falk, and D. A. Lidar, *Suppression of crosstalk in superconducting qubits using dynamical decoupling* (2021), arXiv:2108.04530 [quant-ph].
- [40] A. M. Souza, G. A. Álvarez, and D. Suter, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **370**, 4748 (2012).
- [41] L. Capelluto and T. Alexander, *OpenPulse: Software for Experimental Physicists in Quantum Computing* (2021).
- [42] K. Khodjasteh and D. A. Lidar, *Physical Review Letters* **95**, 180501 (2005).
- [43] G. S. Uhrig, *Phys. Rev. Lett.* **98**, 100504 (2007).
- [44] J. R. West, B. H. Fong, and D. A. Lidar, *Physical Review Letters* **104**, 130501 (2010), arXiv: 0908.4490.
- [45] Z.-Y. Wang and R.-B. Liu, *Phys. Rev. A* **83**, 022306 (2011).
- [46] G. T. Genov, D. Schraft, N. V. Vitanov, and T. Halfmann, *Physical Review Letters* **118**, 133202 (2017).
- [47] E. L. Hahn, *Physical Review* **80**, 580 (1950), publisher: American Physical Society.
- [48] A. A. Maudsley, *Journal of Magnetic Resonance* **69**, 488 (1986).
- [49] H. Carr and E. Purcell, *Phys. Rev.* **94**, 630 (1954).
- [50] S. Meiboom and D. Gill, *Review of Scientific Instruments* **29**, 688 (1958), publisher: American Institute of Physics.
- [51] M. Suzuki, *Communications in Mathematical Physics* **57**, 193 (1977).
- [52] P. Zanardi, *Physics Letters A* **258**, 77 (1999).
- [53] M. S. Byrd and D. A. Lidar, *Quantum Information Processing*

- 1, 19 (2002).
- [54] L. Cywiński, R. M. Lutchyn, C. P. Nave, and S. Das Sarma, *Phys. Rev. B* **77**, 174509 (2008).
- [55] M. J. Biercuk, H. Uys, A. P. VanDevender, N. Shiga, W. M. Itano, and J. J. Bollinger, *Nature* **458**, 996 (2009).
- [56] G. A. Álvarez and D. Suter, *Physical Review Letters* **107**, 230501 (2011), publisher: American Physical Society.
- [57] H. Uys, M. J. Biercuk, and J. J. Bollinger, *Phys. Rev. Lett.* **103**, 040501 (2009).
- [58] G. Quiroz and D. A. Lidar, *Phys. Rev. A* **84**, 042328 (2011).
- [59] G. S. Uhrig and D. A. Lidar, *Physical Review A* **82**, 012301 (2010).
- [60] W.-J. Kuo and D. A. Lidar, *Phys. Rev. A* **84**, 042329 (2011).
- [61] Y. Xia, G. S. Uhrig, and D. A. Lidar, *Phys. Rev. A* **84**, 062332 (2011).
- [62] L. Jiang and A. Imambekov, *Physical Review A* **84**, 060302 (2011).
- [63] W.-J. Kuo, G. Quiroz, G. A. Paz-Silva, and D. A. Lidar, *J. Math. Phys.* **53**, (2012).
- [64] S. Meiboom and D. Gill, *Review of Scientific Instruments* **29**, 688 (1958).
- [65] L. Viola and E. Knill, *Phys. Rev. Lett.* **90**, 037901 (2003).
- [66] G. Quiroz and D. A. Lidar, *Phys. Rev. A* **88**, 052306 (2013).
- [67] A. M. Souza, G. A. Álvarez, and D. Suter, *Physical Review Letters* **106**, 240501 (2011).
- [68] T. Alexander, N. Kanazawa, D. J. Egger, L. Capelluto, C. J. Wood, A. Javadi-Abhari, and D. McKay, *arXiv:2004.06755 [quant-ph]* (2020), arXiv: 2004.06755.
- [69] H. Abraham *et al.*, *Qiskit: An open-source framework for quantum computing* (2019).
- [70] D. C. McKay, T. Alexander, L. Bello, M. J. Biercuk, L. Bishop, J. Chen, J. M. Chow, A. D. Córcoles, D. Egger, S. Filipp, J. Gomez, M. Hush, A. Javadi-Abhari, D. Moreda, P. Nation, B. Paulovicks, E. Winston, C. J. Wood, J. Wootton, and J. M. Gambetta, *arXiv:1809.03452 [quant-ph]* (2018), arXiv: 1809.03452.
- [71] L. Viola, *Journal of Modern Optics* **51**, 2357 (2004).
- [72] S. T. Smith, *M.Sc. Thesis* (2007).
- [73] K. Khodjasteh, D. A. Lidar, and L. Viola, *Phys. Rev. Lett.* **104**, 090501 (2010).
- [74] J.-M. Cai, B. Naydenov, R. Pfeiffer, L. P. McGuinness, K. D. Jahnke, F. Jelezko, M. B. Plenio, and A. Retzker, *New Journal of Physics* **14**, 113023 (2012).
- [75] C. A. Ryan, J. S. Hodges, and D. G. Cory, *Phys. Rev. Lett.* **105**, 200402 (2010).
- [76] R. Freeman, *Spin Choreography: Basic Steps in High Resolution NMR* (Oxford University Press, Oxford, 1998).
- [77] K. R. Brown, A. W. Harrow, and I. L. Chuang, *Physical Review A* **70**, 052318 (2004).
- [78] K. R. Brown, A. W. Harrow, and I. L. Chuang, *Physical Review A* **72**, 039905 (2005).
- [79] IBM Quantum, <https://quantum-computing.ibm.com> (2021).
- [80] K. Khodjasteh and D. A. Lidar, *Phys. Rev. A* **75**, 062310 (2007).
- [81] K. Khodjasteh and D. A. Lidar, *Physical Review A* **68**, 022322 (2003), erratum: *ibid*, *Phys. Rev. A* **72**, 029905 (2005).
- [82] G. S. 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)* (2022) pp. 288–303.
- [83] A. Zlokapa and A. Gheorghiu, *A deep learning model for noise prediction on near-term quantum devices* (2020), *arXiv:2005.10811 [quant-ph]*.
- [84] N. Ezzell, B. Pokharel, L. Tewala, G. Quiroz, and D. A. Lidar, *naezzell/edd: edd arxiv v1.0.2 release* (2022).
- [85] D. C. McKay, C. J. Wood, S. Sheldon, J. M. Chow, and J. M. Gambetta, *Physical Review A* **96**, 022330 (2017), arXiv: 1612.00858.
- [86] U. Haeberlen, *High Resolution NMR in Solids*, Advances in Magnetic Resonance Series, Supplement 1 (Academic Press, New York, 1976).
- [87] L. Viola and E. Knill, *Physical Review A* **68**, 032311 (2003).