

# Demonstration of high-fidelity entangled logical qubits using transmons

---

Received: 8 July 2025

Accepted: 12 February 2026

---

Cite this article as: Vezvae, A., Tripathi, V., Morford-Oberst, M. *et al.* Demonstration of high-fidelity entangled logical qubits using transmons. *Nat Commun* (2026). <https://doi.org/10.1038/s41467-026-70011-3>

---

Arian Vezvae, Vinay Tripathi, Mario Morford-Oberst, Friederike Butt, Victor Kasatkin & Daniel A. Lidar

---

We are providing an unedited version of this manuscript to give early access to its findings. Before final publication, the manuscript will undergo further editing. Please note there may be errors present which affect the content, and all legal disclaimers apply.

If this paper is publishing under a Transparent Peer Review model then Peer Review reports will publish with the final article.

# Demonstration of High-Fidelity Entangled Logical Qubits using Transmons

Arian Vezvaei,<sup>1,2,3,\*</sup> Vinay Tripathi,<sup>2,4,†</sup> Mario Morford-Oberst,<sup>1,2</sup>  
Friederike Butt,<sup>5,6</sup> Victor Kasatkin,<sup>1,2</sup> and Daniel A. Lidar<sup>1,2,3,4,7,‡</sup>

<sup>1</sup>*Department of Electrical & Computer Engineering,  
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>*Quantum Elements, Inc., Thousand Oaks, CA 91361, USA*

<sup>4</sup>*Department of Physics & Astronomy, University of Southern California, Los Angeles, CA 90089, USA*

<sup>5</sup>*Institute for Quantum Information, RWTH Aachen University, Aachen, Germany*

<sup>6</sup>*Institute for Theoretical Nanoelectronics (PGI-2), Forschungszentrum Jülich, Jülich, Germany*

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

## ABSTRACT

Quantum error correction (QEC) codes are necessary to fault-tolerantly operate quantum computers. However, every such code is inherently limited by its inability to detect logical errors. Here, we propose and implement a method that leverages dynamical decoupling (DD) to drastically suppress logical errors. The key to achieving this is to use the normalizer elements of the QEC code as DD pulses, which we refer to as normalizer dynamical decoupling (NDD). The resulting hybrid QEC-NDD strategy is in principle capable of handling arbitrary weight errors. We test an error detecting version of this strategy using IBM transmon devices and the  $[[4, 2, 2]]$  code, demonstrating performance that significantly exceeds the capabilities of using either this code or DD in isolation. We present a method that allows for the detection of logical errors affecting logically encoded Bell states, which, in this case, arise primarily from crosstalk among physical qubits. Building on this, we experimentally demonstrate high-fidelity entangled logical qubits. The fidelities we achieve are beyond-breakeven, i.e., they significantly exceed the corresponding fidelities of unprotected entangled qubits in the same setting.

## INTRODUCTION

Quantum error correction (QEC) [1–3] is fundamental to the realization of fault-tolerant quantum computation [4–6], ensuring the preservation of quantum information undergoing error processes during computation and storage [7, 8]. Numerous successful experimental demonstrations of QEC across various platforms have been reported over the years, with the scale and pace accelerating recently towards genuine fault tolerance [9–16].

Fault-tolerant quantum computing will require the stability of logical qubits on the long timescales of quantum algorithms that solve utility-scale problems [17–19]. Above threshold, on such timescales, low-weight physical errors may transform into logical errors, i.e., errors of weight higher than a deployed fixed-distance QEC code can handle. This can become a concern even below threshold, as long-range spatial and temporal correlations may develop that violate the assumptions underlying fault tolerance theory [20, 21].

Conventionally, suppressing higher-weight errors requires increasing the QEC code distance, e.g., by means of code concatenation [22], or increasingly larger codes such as is done with surface codes [23, 24],

color codes [25], or quantum low-density parity-check codes [26]. Although effective, these strategies result in significant overhead, substantially increasing the number of physical qubits required and the time required to decode and correct errors.

Here, we propose and demonstrate a low-cost method that combines a fixed-distance QEC or quantum error detection (QED) code with dynamical decoupling (DD) [27]. This hybrid method, which we call QEC-NDD in the general case, can handle arbitrary-weight errors. Here, NDD stands for “normalizer dynamical decoupling”; the decoupling sequence is implemented using the normalizer elements of the QEC code as pulses. Throughout, we view NDD as a complement to QEC or QED: it suppresses logical errors that fixed-distance codes cannot detect, and simultaneously lowers the incidence of detectable physical errors. It thereby reduces the discard rate of postselected events.

Standard, “physical” DD, where DD pulses act not at the logical level but at the physical qubit level, has recently shown great progress, improving the fidelity of storing quantum states [28–30], circuits [31, 32], and even the performance of entire algorithms [33, 34]. Furthermore, physical DD can be seamlessly combined with fault-tolerant QEC [35, 36] and several recent QEC experiments have used physical DD profitably [13, 14, 16, 37–39]. However, such pulse sequences can introduce additional errors due to control imperfections in the pulses and due to crosstalk, potentially overshadowing the ben-

\* Corresponding author: vezvaei@usc.edu

† current affiliation: IBM Quantum

‡ lidar@usc.edu

efits of DD. To address this challenge, we design our NDD sequences to be robust against such control errors [40, 41] and crosstalk [42, 43], ensuring that the advantages of logical error suppression are not compromised. This robust design allows our NDD implementation to enhance the protection of the code space as intended, effectively targeting both logical and control errors. We demonstrate the practical utility and advantages of this approach using 127-qubit IBM quantum processors.

The workhorse in our demonstrations is the  $[[4, 2, 2]]$  quantum error detection code, whose two logical qubits we use to prepare logically encoded Bell states. The  $[[4, 2, 2]]$  code is an experimentally relevant primitive with recent demonstrations and proposals in multiple platforms [44–46]. Moreover, its error-detection capability via post-selection lets us seamlessly combine it with DD to isolate and quantify logical-error suppression without the overhead of repeated stabilizer rounds. This makes it an ideal first testbed for normalizer-based decoupling.

To demonstrate the effectiveness of NDD in suppressing logical errors, we first need a method to detect such errors. However, this code is constrained by its low distance and is incapable of detecting prevalent logical errors, including  $ZZ$  crosstalk errors. To overcome this limitation, we design experiments in which, through the use of logically encoded Bell states, we nevertheless unequivocally detect the occurrence of logical errors for a known input logical state. This is then followed by the implementation of various versions of NDD, each employing different subsets of normalizer elements to construct the sequence. These implementations effectively demonstrate significant suppression of logical errors and substantial improvement in the fidelity of the code space.

## RESULTS

**Error model.** We consider the most general “total decoherence” system-bath interaction for  $n$  qubits:

$$H_{SB} = \sum_{j=1}^{4^n-1} \alpha_j P_j \otimes B_j. \quad (1)$$

In writing this expression we exclude the bath Hamiltonian  $I_S \otimes B_0$  (where  $B_0 = H_B$ ) and select exactly one representative  $P_j$  from each of the  $4^n$  equivalence classes  $\tilde{\mathcal{P}}_n$  such that (s.t.)  $P_0 = I_S$  (we use the notation  $I_S$  to denote the identity operator on the entire system Hilbert space; we use  $I$  for a generic identity operator). There is still some freedom in how we order the elements of  $\tilde{\mathcal{P}}_n$  and which phases we choose. We use this freedom later to organize  $\{P_j\}_{j=1}^{4^n-1}$  in the context of  $[[n, k, d]]$  codes.

Our model for  $H_{SB}$  is general enough to contain system-only error terms, since any of the bath operators  $B_j$  may be set equal to the bath identity operator  $I_B$ . Henceforth, when unambiguous, we use  $I$  to denote the identity operator regardless of the space on which it acts or its dimension. The joint system-bath free-evolution

unitary is  $f_\tau = e^{-i\tau(H_{SB} + I \otimes H_B)}$ . The adjective “free” refers to the fact that we are leaving out any system-only control Hamiltonian; such control will enter later through DD or QEC. The goal of the latter is, respectively, to suppress or correct the deleterious effects caused by  $H_{SB}$ .

**Dynamical decoupling and quantum error correction.** In its simplest form, accounting only for first-order decoupling and ignoring pulse errors, DD theory can be understood as group symmetrization [47]. Let  $U(m)$  denote the group of unitary operators on  $\mathbb{C}^m$ . Consider a discrete group  $\mathcal{G} \subset U(2^n)$  with elements  $\{g_j\}_{j=0}^{|\mathcal{G}|-1}$  representing unitary transformations  $g_j$  acting purely on the system. We set  $g_0 \equiv g_{|\mathcal{G}|} \equiv I_S$  and refer to the corresponding phase-stripped group  $\tilde{\mathcal{G}} = \pi(\mathcal{G}) = \{\tilde{g}_j\}_{j=0}^{K-1} \subset U(2^n)/U(1)$  as the “decoupling group”, where  $K \equiv |\tilde{\mathcal{G}}| = |\mathcal{G}|/4$ . This yields the DD cycle

$$U(T) = \prod_{j=0}^{K-1} \tilde{g}_j^\dagger f_\tau \tilde{g}_j = e^{-iT(\langle H_{SB} \rangle_{\mathcal{G}} + I \otimes H_B)} + O(T^2) \quad (2)$$

$$= p_K f_\tau p_{K-1} f_\tau \cdots f_\tau p_1 f_\tau.$$

Here,  $\{p_j = \tilde{g}_j \tilde{g}_{j-1}^\dagger\}_{j=1}^K$  are the pulses of the corresponding DD pulse sequence,  $\tau$  is the pulse interval, and  $T = K\tau$  is the total duration of one repetition of the sequence. Note that, equivalently,  $\tilde{g}_{j-1} = p_{j-1} \cdots p_0$ , where  $p_0 \equiv I_S$ . We set  $\hbar = 1$  throughout, so  $\tau$  is in units of time ( $\mu$ s in our experiments) and all Hamiltonians have inverse-time units.

$\langle H_{SB} \rangle_{\mathcal{G}}$  is the group-symmetrized system-bath Hamiltonian, where

$$\langle A \rangle_{\mathcal{G}} \equiv \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} g^\dagger A g = \frac{1}{K} \sum_{\tilde{g} \in \tilde{\mathcal{G}}} \tilde{g}^\dagger A \tilde{g} \quad (3)$$

is the projection of the arbitrary bounded system-bath operator  $A$  onto the subalgebra of operators that commute with every element of  $\mathcal{G}$  (i.e., its commutant).  $\langle A \rangle_{\mathcal{G}}$  is also called the group average of  $A$  with respect to  $\mathcal{G}$ .

*Definition 1* (Decoupling). Let  $A$  be an arbitrary operator acting on the joint system-bath Hilbert space.  $\tilde{\mathcal{G}}$  decouples  $A$  (to first order) if  $\langle A \rangle_{\mathcal{G}} = cI \otimes B$  where  $c$  is a constant, including zero, and  $B$  is an arbitrary bath operator.

To set up an example, let us define  $\mathcal{P}_n = \{\pm P_j, \pm iP_j\}_{j=0}^{4^n-1} \subset U(2^n)$  to be the full Pauli group with  $4^{n+1}$  elements generated by  $\{I, X, Y, Z\}^{\otimes n}$  and the phase factors  $\{\pm 1, \pm i\}$ , where  $X \equiv \sigma^x$  and likewise for  $Y$  and  $Z$ . There is a natural projection (quotient homomorphism)  $\pi: \mathcal{P}_n \rightarrow \tilde{\mathcal{P}}_n$  which takes the Pauli operator and ignores its global phase, yielding a “phase-stripped” Pauli. Correspondingly, let  $\tilde{\mathcal{P}}_n = \mathcal{P}_n / \{\pm 1, \pm i\} \subset U(2^n)/U(1)$  be the Abelian factor group with  $4^n$  elements, also known as the projective Pauli group.

If we pick,  $\tilde{\mathcal{G}} = \tilde{\mathcal{P}}_n$ , then  $\tilde{\mathcal{G}}$  decouples an arbitrary  $n$ -qubit system-bath Hamiltonian since then  $\langle H_{SB} \rangle_{\mathcal{G}} = 0$ ,

albeit at a sequence time cost of  $T = 4^n \tau$  [27]. To illustrate this, consider  $n = 1$ ; the most general system-bath Hamiltonian of a single qubit is  $H_{SB} = \sum_{i=1}^3 \alpha_i P_i \otimes B_i$  with  $P_i \in \mathcal{P}_1 \setminus I$ . If  $\tilde{\mathcal{G}} = \tilde{\mathcal{P}}_1 = \langle X, Z \rangle = \{I, X, Y, Z\}$ , then  $U(T)$  simplifies into the well-known (universal) XY4 sequence  $U(4\tau) = Y f_\tau X f_\tau Y f_\tau X f_\tau$  [48], and indeed,  $\langle H_{SB} \rangle_{\mathcal{G}} = 0$ . The XX sequence is  $U(2\tau) = X f_\tau X f_\tau$ , and decouples  $Y$  and  $Z$ , but not  $X$  errors. The XY4 sequence can be seen as a concatenation of the XX and ZZ sequences, with the latter defined similarly to XX [49].

Note that a common misconception is that DD is not effective against purely Markovian noise; however, this is not the case, essentially since even in the Markovian limit, the bath can have a nonzero correlation time [50–53].

In general, decoupling using subgroups of  $\tilde{\mathcal{P}}_n$  will eliminate parts of  $H_{SB}$ , presenting an opportunity to selectively combine DD with QEC [35, 54]. See Methods for pertinent background on QEC codes.

Let  $\mathcal{S}$  be a stabilizer group specifying an  $[[n, k, d]]$  stabilizer code. The group of logical operators is  $\mathcal{N}(\mathcal{S})/\mathcal{S}$ , where  $\mathcal{N}(\mathcal{S})$  is the normalizer of  $\mathcal{S}$  in  $\mathcal{P}_n$  (which here coincides with the centralizer, i.e., the group of Pauli operators commuting with all elements of  $\mathcal{S}$ ). We can pick a canonical set of generators of the stabilizer group and a canonical group of logical operators  $\mathcal{L}$ . (That is, any transversal for  $\mathcal{S}$  in  $\mathcal{N}(\mathcal{S})$  satisfying  $\pi^{-1}(\pi(\mathcal{L})) = \mathcal{L}$  can be picked as  $\mathcal{L}$ , i.e., any subgroup of  $\mathcal{N}(\mathcal{S})$  with exactly one representative from each coset  $LS \subset \mathcal{N}(\mathcal{S})$  s.t.  $iI \in \mathcal{L}$ .) Any such  $\mathcal{L}$  is isomorphic to  $\mathcal{P}_k$  and can be thought of as a group of Paulis acting on logical qubits.

We can use the remaining freedom mentioned above to reorder  $\{P_j\}_{j=0}^{4^n-1}$  and pick the phases of those operators to ensure that

- $P_0 = I_S$
- $\mathcal{S}_* = \{P_1, \dots, P_{2^{n-k}-1}\} = \mathcal{S} \setminus \{I_S\}$  (these are non-identity stabilizers acting trivially on the code space)
- $\mathcal{L}_* = \{P_{2^{n-k}}, \dots, P_{2^{n+k}-1}\} \subset \mathcal{N}(\mathcal{S}) \setminus \pi^{-1}(\tilde{\mathcal{S}})$  (these manifest as undetectable logical errors when acting on the code space)
- $\mathcal{D}_* = \{P_{2^{n+k}}, \dots, P_{4^n-1}\} \subset \mathcal{P}_n \setminus \mathcal{N}(\mathcal{S})$  (these are detectable errors).

We chose the phases of the representatives of  $\tilde{\mathcal{S}}$  to ensure they are actually in  $\mathcal{S}$  (and not different from elements of  $\mathcal{S}$  by a phase). This represents  $\{P_j\}_{j=0}^{4^n-1}$  as a disjoint union

$$\{P_j\}_{j=0}^{4^n-1} = \{I_S\} \dot{\cup} \mathcal{S}_* \dot{\cup} \mathcal{L}_* \dot{\cup} \mathcal{D}_*. \quad (4)$$

The system Hilbert space  $\mathcal{H}$  can be split as

$$\mathcal{H} = \mathcal{H}_{\text{stab}} \otimes \mathcal{H}_{\text{log}}, \quad (5)$$

where  $\mathcal{S}$  acts on the first component and  $\mathcal{L}$  acts on the second, and where  $\dim(\mathcal{H}_{\text{stab}}) = 2^{n-k}$  and  $\dim(\mathcal{H}_{\text{log}}) =$

$2^k$ , corresponding to the  $n-k$  (virtual) syndrome qubits and  $k$  logical qubits, respectively.

With respect to this virtual tensor product decomposition induced by the syndrome measurement [55], every Pauli operator  $E \in \mathcal{S}_*$ ,  $\mathcal{L}_*$ , or  $\mathcal{D}_*$  can be written as:

- $E = P \otimes I \in \mathcal{S}_*$  with  $P \in \mathcal{P}_{n-k}$  (a stabilizer with trivial action on the code space)
- $E = P \otimes Q \in \mathcal{L}_*$  with  $[P, \mathcal{S}_*] = 0$  (a logical operator manifesting as an error if it appears in  $H_{SB}$ ).
- $E = P \otimes Q \in \mathcal{D}_*$  with  $\{P, S_i\} = 0$  for one or more  $S_i \in \mathcal{S}_*$  (a detectable error).

Correspondingly, it is always possible to write the system-bath interaction [Eq. (1)] as

$$H_{SB} = H_{SB}^{\mathcal{S}_*} + H_{SB}^{\mathcal{L}_*} + H_{SB}^{\mathcal{D}_*} \quad (6a)$$

$$H_{SB}^{\mathcal{S}_*} = \sum_{i=1}^{2^{n-k}-1} S_i \otimes B_i^{\text{stab}}, \quad S_i \in \mathcal{S}_* \quad (6b)$$

$$H_{SB}^{\mathcal{L}_*} = \sum_{i=1}^{2^{n-k}(4^k-1)} L_i \otimes B_i^{\text{log}}, \quad L_i \in \mathcal{L}_* \quad (6c)$$

$$H_{SB}^{\mathcal{D}_*} = \sum_{i=1}^{4^n-2^{n+k}} D_i \otimes B_i^{\text{det}}, \quad D_i \in \mathcal{D}_*, \quad (6d)$$

where  $\{B_i^{\text{stab}}, B_i^{\text{log}}, B_i^{\text{det}}\}$  are bath operators with units of inverse time (with  $\hbar = 1$ ).

Eq. (6) groups terms with similar action on the code space  $\mathcal{C}$ . In total,  $H_{SB} + I_S \otimes H_B$  has  $4^n$  linearly independent terms corresponding to the elements of  $\tilde{\mathcal{P}}_n$ . Specifically,  $H_{SB}^{\mathcal{S}_*} + I_S \otimes H_B$  collects all  $2^{n-k}$  distinct terms in  $\{I_S\} \dot{\cup} \mathcal{S}_*$  that act as the identity operator on the code space and, therefore, are harmless.  $H_{SB}^{\mathcal{L}_*}$  collects all  $2^{n-k}(4^k-1)$  Pauli operators in  $\mathcal{L}_*$ ; these terms are non-trivial logical operators that leave the code space (and each of the other syndrome spaces) invariant. Finally,  $H_{SB}^{\mathcal{D}_*}$  collects all remaining  $4^n - 2^{n-k} - 2^{n-k}(4^k-1) = 4^n - 2^{n+k}$  terms in  $\mathcal{D}_*$ , which correspond to detectable and potentially correctable errors.

We can now establish when an error  $E \in \mathcal{P}_n$  is decoupled.

*Lemma 1.* Let  $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$  be the Hilbert space of  $n$  qubits. Then all elements of  $\mathcal{P}_n$  are unitary operators on  $\mathcal{H}$ . Consider a term in  $H_{SB}$  corresponding to some  $E \in \mathcal{P}_n$ . It is decoupled by  $\tilde{\mathcal{G}}$  in the sense of Definition 1 if and only if there exists  $g \in \mathcal{G}$  anticommuting with  $E$ .

*Proof.* If there exists  $g \in \mathcal{G}$  that anticommutes with  $E$  then  $g^\dagger E g = -E$  and

$$\langle E \rangle_{\mathcal{G}} = \frac{1}{|\mathcal{G}|} \sum_{h \in \mathcal{G}} h^\dagger E h = -\frac{1}{|\mathcal{G}|} \sum_{h \in \mathcal{G}} (gh)^\dagger E gh = -\langle E \rangle_{\mathcal{G}}, \quad (7)$$

so the group average  $\langle E \rangle_{\mathcal{G}} = 0$ . If there is no such  $g$ , then all elements of  $\mathcal{G}$  commute with  $E$  (because every two

elements of  $\mathcal{P}_n$  either commute or anticommute). Then  $g^\dagger E g = E$  for all  $g \in \mathcal{G}$ , hence  $\langle E \rangle_{\mathcal{G}} = E$  (i.e., the term is undecoupled).

**Encoded dynamical decoupling.** A key observation is that the decoupling group is arbitrary [56, 57] and may, in particular, be formed from the stabilizer group  $\mathcal{S}$  and/or the group of logical operators  $\mathcal{L}$  of a stabilizer code  $\mathcal{C}$  [36]. We call these different choices ‘‘encoded dynamical decoupling,’’ as they all involve the use of encoded operations (relative to a QEC code) as DD pulses, instead of physical DD pulses. We next provide several examples of encoded DD groups.

When using a QEC code, the approach of using the full Pauli group  $\mathcal{P}_n$  is excessive, since there is no need to decouple the terms in  $H_{SB}^{\mathcal{S}}$ ; this observation motivated Ref. [36] to introduce SLDD, which achieves a reduction from a DD group order of  $4^n$  to  $2^{n+k}$ . The savings are substantial when  $k \ll n$ .

*Definition 2* (SLDD). For any stabilizer group  $\mathcal{S}$  and associated canonical group of logical operators  $\mathcal{L}$ , the stabilizer-logical dynamical decoupling (SLDD) group is

$$\tilde{\mathcal{G}}_{\text{SLDD}} = \pi(\mathcal{S}\mathcal{L}). \quad (8)$$

Note that  $\mathcal{S}\mathcal{L} = \mathcal{N}(\mathcal{S})$  and thus independent of the choice of  $\mathcal{L}$ .

The main weakness of SLDD is that it still incurs a cost that is exponential in the number of physical qubits  $n$ . This observation motivates us to consider a different hybrid approach that shifts more of the burden for handling correctable errors to the QEC code and uses only the logical operators of the code as DD pulses:

*Definition 3* (LDD). Pick a canonical group of logical operators  $\mathcal{L}$ . The logical dynamical decoupling (LDD) group is

$$\tilde{\mathcal{G}}_{\text{LDD}} \equiv \tilde{\mathcal{L}} = \pi(\mathcal{L}). \quad (9)$$

Compared to SLDD, this approach has the desired effect of reducing the DD group order to  $T \leq 4^k$ , independent of  $n$ .

For the sake of completeness, we also mention the case of DD using only the stabilizers:

*Definition 4* (SDD). For any stabilizer group  $\mathcal{S}$ , the SDD group is

$$\tilde{\mathcal{G}}_{\text{SDD}} \equiv \tilde{\mathcal{S}} = \pi(\mathcal{S}). \quad (10)$$

SDD can be seen as an alternative to QEC since it decouples all detectable errors. However, it does not decouple any logical errors and is therefore not a particularly useful method when used in conjunction with error correction. It does play an important role in scenarios where QEC is not viable, such as in adiabatic quantum computing [58].

We can now classify the different decoupling groups by the errors they decouple.

*Theorem 1* (Decoupling by group type).

1. SDD decouples all detectable errors:  $\langle H_{SB}^{\mathcal{D}_*} \rangle_{\tilde{\mathcal{G}}_{\text{SDD}}} = 0$ .
2. LDD decouples all logical errors:  $\langle H_{SB}^{\mathcal{L}_*} \rangle_{\tilde{\mathcal{G}}_{\text{LDD}}} = 0$ .
3. SLDD decouples all errors:  $\langle H_{SB}^{\mathcal{L}_*} + H_{SB}^{\mathcal{D}_*} \rangle_{\tilde{\mathcal{G}}_{\text{SLDD}}} = 0$ .

*Proof.* It follows from Lemma 1 that it suffices to show that for each term in  $H_{SB}^{\mathcal{S}_*}$ ,  $H_{SB}^{\mathcal{L}_*}$ , or  $H_{SB}^{\mathcal{D}_*}$ , there exists  $g \in \mathcal{G}$  it anticommutes with, where  $\mathcal{G}$  is the corresponding decoupling group. Indeed:

1. SDD: Each detectable error anticommutes with at least one element of  $\mathcal{S}$ .
2. LDD: Each nontrivial logical error anticommutes with at least one element of  $\mathcal{L}$ .
3. Since  $I \in \mathcal{L}$ , the stabilizer  $\mathcal{S} \subset \mathcal{G}_{\text{SLDD}} = \{SL \mid L \in \mathcal{L}\}$ . Thus, the group average of  $H_{SB}^{\mathcal{L}_*} + H_{SB}^{\mathcal{D}_*}$  with respect to  $\mathcal{G}_{\text{SLDD}}$  includes both  $\mathcal{S}$ , which decouples all detectable errors, and  $\mathcal{L}$ , which decouples all logical errors.

Table I summarizes the relative resource cost of full Pauli group DD, SDD, LDD, and SLDD.

Comparative DD resource cost			
Decoupling group $\mathcal{G}$	# of pulses	error type suppressed	requires QEC
$\tilde{\mathcal{P}}_n$	$2^{2n}$	all	No
SDD	$2^{n-k}$	detectable	No
LDD	$2^{2k}$	logical	Yes
SLDD	$2^{n+k}$	detectable & logical	No
NDD	$2^r$	logical, some physical	yes

**Table I. Comparative resource cost for various DD methods.** Number of pulses required to suppress the most general system-bath Hamiltonian  $H_{SB}$  [Eq. (6)] acting on  $n$  qubits, and whether active error correction is required in addition. Full Pauli group DD and SLDD suppress all error mechanisms that can decohere code states, while LDD does not, as it leaves dealing with these error mechanisms for the QEC cycle. Consequently, LDD does not scale with  $n$  but with  $k$ , whereas Pauli group DD and SLDD both scale with  $n$ . In the NDD case,  $r$  is the number of chosen normalizer generators.

It is common to use only subgroups of the complete group of logical operators. In the context of standard (unencoded) DD, this is the case, e.g., when using single-axis sequences such as XX instead of the universal XY4 sequence. In the present context, we would like to consider subgroups of  $\mathcal{N}(\mathcal{S})$  other than  $\mathcal{S}$  and  $\mathcal{L}$ , which will

similarly suppress selected subsets of errors. We refer to this as normalizer DD, since the corresponding DD group elements are generic normalizer elements, as opposed to canonical logical operators.

*Definition 5* (NDD). Normalizer DD (NDD) is any pulse sequence formed from using a subgroup of  $\tilde{\mathcal{G}}_{\text{SLDD}}$  other than  $\tilde{\mathcal{G}}_{\text{SDD}}$  or  $\tilde{\mathcal{G}}_{\text{LDD}}$ .

NDD is our main focus experimentally. Unlike SLDD or LDD, the order of an NDD sequence is set by the number  $r$  of chosen normalizer generators, not by  $n$  or  $k$ . Any  $r$  elements that anticommute with the targeted error set yield a first-order sequence of length  $K = 2^r$  (e.g.,  $r = 2$  yields a four-pulse XY4), i.e., a constant-length sequence independent of code size.

This resource flexibility lets one tailor NDD to both the code and the hardware by cancelling only the dominant channels (e.g., ZZ-related logical Z errors), while avoiding hardware-problematic controls such as virtual-Z pulses on fixed-frequency transmons [59]. In practice, this keeps sequences short and robust and is what enables the present experimental realization.

Because the chosen normalizer generators commute with a standard subset of logical Clifford operations, NDD can be interleaved with such gates without breaking first-order decoupling; in our implementation, the encoding uses logical SWAP-based CNOTs while gaps are protected by physical DD and the code space by NDD.

It is important to emphasize that additional errors may and will generally be decoupled by each of the decoupling groups, beyond those mentioned in Theorem 1. For example, a detectable weight-1 Pauli operator that anticommutes with an element of  $\mathcal{G}_{\text{LDD}}$  will be decoupled despite not being a logical error for any code with  $d \geq 2$ .

Another important point is that since we are employing a Hamiltonian noise model, any terms that arise in second order perturbation theory, i.e., the  $O(T^2)$  terms we are neglecting in Eq. (2), will mix  $H_{SB}^{S_B}$ ,  $H_{SB}^{L_B}$ , and  $H_{SB}^{D_B}$  through commutators. This means that the clean separation we have assumed between error types disappears at this higher order of perturbation theory. However, higher-order decoupling sequences are known that achieve suppression up to arbitrary order  $q$  [i.e., leaving an  $O(T^{q+1})$  unsuppressed error term in  $U(T)$ ] [49, 60, 61]. For simplicity, in this work we restrict our attention to first-order sequences.

#### Illustration of LDD and NDD: the $[[4, 2, 2]]$ code.

To illustrate the concepts discussed above, we analyze the example of the  $[[4, 2, 2]]$  code, as it plays a key role in our experiments. The stabilizer group of the  $[[4, 2, 2]]$  code is  $\mathcal{S} = \langle XXXX, ZZZZ \rangle$ . A canonical set of logical operators for the code can be chosen such that  $\bar{X}_1 = XIIX$ ,  $\bar{X}_2 = IIXX$ ,  $\bar{Z}_1 = IIZZ$  and  $\bar{Z}_2 = ZIIZ$ , i.e.,  $\mathcal{L} = \langle i, XIIX, IIXX, IIZZ, ZIIZ \rangle$  (see Methods for additional pertinent details regarding this code). Thus,  $\{XIIX, IIXX, IIZZ, ZIIZ\}$  is the generator set of the 16-element LDD group  $\tilde{\mathcal{G}}_{\text{LDD}} = \{\tilde{g}_j\}$ . This LDD group (as well as any of the corresponding NDD groups) sup-

presses every logical error  $E_{\text{log}}$  in the  $2^{n-k}(4^k - 1) = 4 \times 15 = 60$ -element group  $(\mathcal{N}(\mathcal{S}) \setminus \mathcal{S})/\{\pm 1, \pm i\}$ : the group-averaged logical system-bath Hamiltonian  $\langle H_{SB}^{L_B} \rangle_{\mathcal{G}}$  [Eq. (6c)] vanishes, as every one of its terms is a logical operator that anticommutes with an element of  $\mathcal{G}$ .

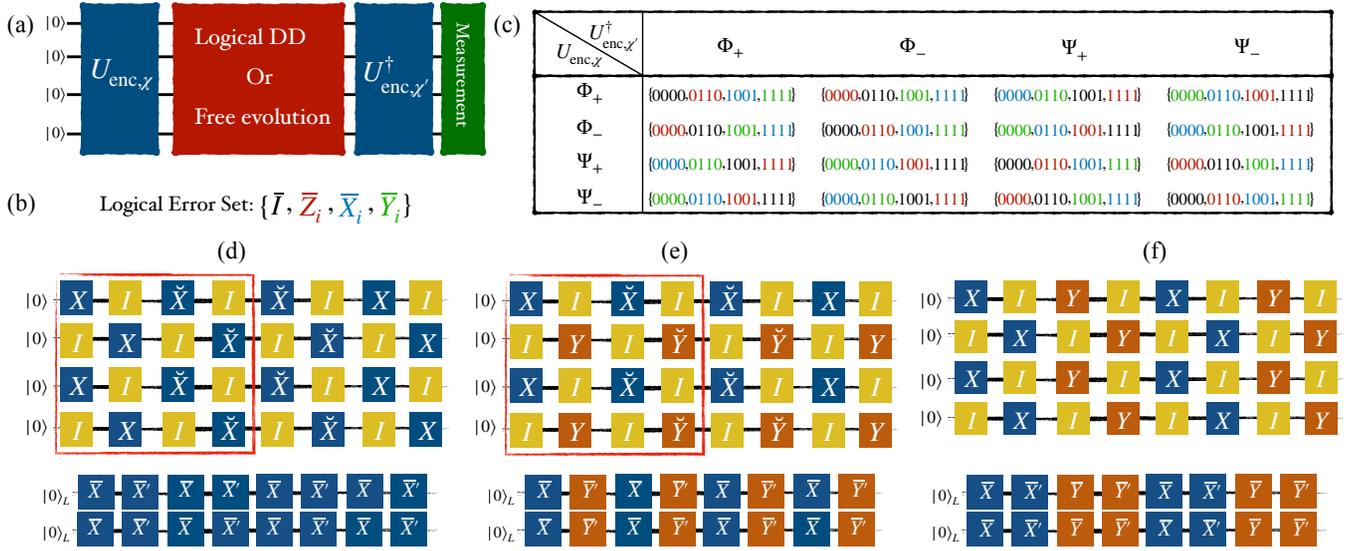
This leaves  $4^n - 2^{n+k} = 256 - 64 = 192$  detectable errors. These errors can be written compactly as  $\mathcal{D} = \{(XIII \cdot \tilde{\mathcal{N}}(\mathcal{S})) \cup (YIII \cdot \tilde{\mathcal{N}}(\mathcal{S})) \cup (ZIII \cdot \tilde{\mathcal{N}}(\mathcal{S}))\}$ , yielding  $3 \times |\tilde{\mathcal{N}}(\mathcal{S})| = 192$  terms. Each such error is detectable because it anticommutes with at least one element of  $\mathcal{S}$ . Of these, only the 12 errors of the form  $PQPP$ , with  $P \neq Q \in \{I, X, Y, Z\}$ , are unsuppressed by LDD.

Now suppose that instead of suppressing the full  $H_{SB}^{L_B}$ , the goal is to suppress just  $H_{SB}^{\bar{Z}} \equiv \bar{Z}_1 \otimes B_1^{\text{log}} + \bar{Z}_2 \otimes B_2^{\text{log}}$ . This scenario is particularly relevant to the experiments we describe below. In this case, it suffices to use the smaller LDD-subgroup  $\langle \bar{X}_1 \bar{X}_2 \rangle$ , since both  $\bar{Z}_1$  and  $\bar{Z}_2$  anticommute with  $\bar{X}\bar{X} \equiv \bar{X}_1 \bar{X}_2$ . The corresponding LDD cycle is then  $\bar{X}\bar{X} f_{\tau} \bar{X}\bar{X} f_{\tau}$ , where  $f_{\tau} = \exp(-i\tau H_{SB}^{\bar{Z}})$ . For crosstalk-related reasons that will become apparent below, it turns out that it is advantageous to replace the second  $\bar{X}\bar{X}$  by  $S_1 \bar{X}\bar{X} = IXIX \equiv \bar{X}'\bar{X}'$ , where  $S_1 = XXXX \in \mathcal{S}$ . This replacement yields the 4-pulse NDD cycle  $\bar{X}'\bar{X}' f_{\tau} \bar{X}\bar{X} f_{\tau} \bar{X}'\bar{X}' f_{\tau} \bar{X}\bar{X} f_{\tau} = (\bar{X}'\bar{X}' f_{\tau} \bar{X}'\bar{X}') (S_1 f_{\tau} S_1) (\bar{X}\bar{X} f_{\tau} \bar{X}\bar{X}) (I f_{\tau} I)$ . This cycle is generated by the NDD group  $\tilde{\mathcal{G}} = \langle \bar{X}\bar{X}, S_1 \rangle = \{I, \bar{X}\bar{X}, S_1, \bar{X}'\bar{X}'\}$ . Below, we refer to this decoupling group as NXX.

As another example of NDD, consider the encoded XY4 sequence with the LDD group  $\tilde{\mathcal{G}} = \langle \bar{X}\bar{X}, \bar{Z}\bar{Z} \rangle$  and DD cycle  $\bar{Y}\bar{Y} f_{\tau} \bar{X}\bar{X} f_{\tau} \bar{Y}\bar{Y} f_{\tau} \bar{X}\bar{X} f_{\tau}$ . Again, for crosstalk-related reasons it will turn out to be advantageous to replace the canonical  $\bar{Y}\bar{Y} = YIYI$  by  $\bar{Y}'\bar{Y}' = S_2 \bar{Y}\bar{Y}$ , where  $S_2 = YYYY \in \mathcal{S}$ . The resulting DD cycle  $\bar{Y}'\bar{Y}' f_{\tau} \bar{X}\bar{X} f_{\tau} \bar{Y}'\bar{Y}' f_{\tau} \bar{X}\bar{X} f_{\tau}$  is generated by the NDD group  $\tilde{\mathcal{G}} = \langle \bar{X}\bar{X}, \bar{Z}'\bar{Z}' \rangle = \{I, \bar{X}\bar{X}, \bar{Y}'\bar{Y}', \bar{Z}'\bar{Z}'\}$ , where  $\bar{Z}'\bar{Z}' = S_2 \bar{Z}\bar{Z}$ . From here on, we refer to this decoupling sequence as NXY4, which is capable of suppressing all types of single logical qubit errors.

**Experimental design.** For our experiments, while sequences utilizing the full LDD/NDD group can be constructed (see Methods), we instead use a subgroup-NDD, generated by a subgroup of  $\mathcal{L}$  and a subgroup of  $\mathcal{S}$ . This subgroup NDD works better in practice because of its robustness and ability to suppress crosstalk. This section is dedicated to demonstrating experimentally how our hybrid QED + subgroup-NDD protocol results in a significant fidelity enhancement of entangled logical qubits.

The data for the experiments we report here were collected on two separate occasions from a total of 24 sets of four-qubit experiments run on the `ibm_kyiv` quantum processor. Dataset 1, using 14 sets of qubits, was collected during the week of August 12<sup>th</sup> (2024). Dataset 2, using 9 sets of qubits, was collected on March 1<sup>st</sup> and 2<sup>nd</sup> (2025). These 9 four-qubit sets differ from those in dataset 1 in order to test the robustness of our results. The figure captions below specify the dataset. For each



**Figure 1. Main Setup.** (a) Schematic of the experimental design: we encode a logical Bell state  $\chi$  using the  $[[4, 2, 2]]$  code and let it undergo either free evolution or NDD for a duration  $\tau$ . We then run the unencoding circuit for  $\chi'$  and measure the resulting bitstring. The encoding and unencoding circuits together take up to  $5\mu\text{s}$ . (b) The logical error set that relates the four Bell states as a result of a logical operator acting on one ( $i = 1$  or  $2$ ) of the two logical qubits. (c) Interpretation of different bitstrings for an encoded state ( $U_{\text{enc},\chi}$ ), depending on the unencoding ( $U_{\text{enc},\chi'}^\dagger$ ) circuit used. The color code indicates that the encoded state has undergone either no error (black) or one of the logical errors (red, blue, green) indicated in (b). The DD sequences used in our experiments are (d) Robust Normalizer XX (RNXX), (e) Robust Normalizer XY4 (RNXY4), and (f) physically Staggered XY4 (SXY4). Each panel shows both the physical implementation (top) and the logical interpretation of each sequence (bottom). A dagger in (d) and (e) denotes a negative  $\pi$  rotation, i.e.,  $X = R_x(\pi) = \exp(-i\pi\sigma^x/2)$ ,  $\check{X} = R_x(-\pi) = \exp(i\pi\sigma^x/2)$ , and likewise for  $Y$ . In (d) the first column is the canonical  $\bar{X}\bar{X} = XIXI$  (see [Methods](#)), the second column is  $\bar{X}'\bar{X}' = S_1\bar{X}\bar{X}$ , where  $S_1 = XXXX \in \mathcal{S}$ , etc. Likewise, in (e), the second column is  $\bar{Y}'\bar{Y}' = S_2\bar{Y}\bar{Y}$ , where the canonical  $\bar{Y}\bar{Y} = YIYI$  and  $S_2 = YYYY \in \mathcal{S}$ . On the  $[[4, 2, 2]]$  code, staggered physical XY4 shown in (f) acts as a logical  $Z$ -only NDD (bottom), hence comparisons between SXY4 and NDD reflect logical-level decoupling of the same dominant channel.

data point in our results, fidelity is independently calculated for each qubit set and then bootstrapped by re-sampling. The mean fidelity and standard deviation are derived from the bootstrapped data, where larger error bars indicate greater variability among the qubit sets.

The `ibm_kyiv` processor consists of coupled, fixed-frequency transmons. Such qubits exhibit an always-on interaction between adjacent pairs [\[62\]](#). This crosstalk gives rise to weight-2 error terms that correspond to logical errors with our  $[[4, 2, 2]]$  code choice. Other errors of weight  $\geq 2$ , if present, would likewise correspond to logical errors. To demonstrate that NDD can suppress all logical errors, we design an experiment that enables the unequivocal detection of such errors.

To this end, we use the two logical qubits of the  $[[4, 2, 2]]$  code to create logical Bell states  $|\Phi_\pm\rangle = (|\bar{00}\rangle \pm |\bar{11}\rangle)/\sqrt{2}$  and  $|\Psi_\pm\rangle = (|\bar{01}\rangle \pm |\bar{10}\rangle)/\sqrt{2}$ . The corresponding encoding circuits  $U_{\text{enc},\chi}$ , where  $\chi \in \{|\Phi_\pm\rangle, |\Psi_\pm\rangle\}$ , create two copies of physical Bell states, followed by the application of a logical controlled-NOT ( $\overline{\text{CNOT}}$ ) to generate the logical Bell states (see [Methods](#)). In principle, it would be possible to detect specific logical errors using other logical states (e.g.,  $|\bar{00}\rangle$  would allow us to detect  $X$ - or  $Y$ -type logical errors), but we specifically opt for logical Bell states as they have shorter encoding cir-

cuits. Additionally, the four logical Bell states form an orthonormal basis and the logical errors we are interested in permute this basis.

To estimate the fidelity of a logical Bell state  $\chi$ , we start from the physical ground state  $|0000\rangle$ , encode into the state of interest by applying  $U_{\text{enc},\chi}$ , and then let it evolve for some time  $\tau$ , either freely or subject to DD. After time  $\tau$  we unencode by applying  $U_{\text{enc},\chi'}^\dagger$ , and measure all qubits. Measuring the bitstring  $0000$  would indicate that no error occurred.

If, instead, we obtain any of the odd-Hamming weight bitstrings, this would signal detection of a physical error, as such states are not in the codespace. The final possibility is that we obtain one of the other even-Hamming weight bitstrings  $\{0110, 1001, 1111\}$ , corresponding to a logical error. This procedure can be used to estimate the probability of finding the ground state as the empirical fraction of ground state measurement outcomes, which is also the fidelity of the logical Bell state  $\chi$ .

To obtain the fidelity of a logical error, we first observe that each of the logical Bell states is related to the other three logical Bell states through a specific logical error:  $\chi' = \bar{O}\chi$ , for  $\bar{O} \in \{\bar{X}_i, \bar{Y}_i, \bar{Z}_i\}$  and  $i = 1$  or  $2$ ; i.e., a logical operator that acts on either of the two logical qubits. More generally, any other combinations of logical

	No DD	XY4	UR <sub>6</sub>	UR <sub>8</sub>	UR <sub>10</sub>	UR <sub>18</sub>	RGAs <sub>a</sub>
$ \Phi_+\rangle$	65.44 ± 0.24	83.94 ± 0.18	83.08 ± 0.19	83.11 ± 0.19	82.05 ± 0.19	81.43 ± 0.20	84.20 ± 0.18
$ \Phi_-\rangle$	61.23 ± 0.24	83.78 ± 0.18	81.92 ± 0.19	82.63 ± 0.19	81.07 ± 0.19	80.81 ± 0.18	83.34 ± 0.18
$ \Psi_+\rangle$	66.18 ± 0.23	84.68 ± 0.18	84.17 ± 0.18	83.37 ± 0.19	82.46 ± 0.19	81.21 ± 0.19	84.65 ± 0.18
$ \Psi_-\rangle$	65.43 ± 0.23	86.57 ± 0.17	85.20 ± 0.18	85.53 ± 0.18	84.17 ± 0.18	83.44 ± 0.17	86.38 ± 0.17

Table II. **Fidelity improvement (%) of the encoded logical Bell states with physical circuit-DD.** Experiments were performed on fourteen different sets of qubits on `ibm_kyiv`, with 4000 shots per set. The fidelities were computed independently for each set and then bootstrapped. The reported values represent the bootstrapped averages with  $1\sigma$  uncertainty. The boxed values are the highest fidelities in each row.

weight-1 errors also map the input Bell state to the other Bell states. That is, for  $\bar{p}, \bar{q} \in \{\bar{X}, \bar{Y}, \bar{Z}\}$ , we have  $\chi' = (\bar{p} \otimes \bar{q})\chi = I \otimes (\bar{p}\bar{q})\chi = (\bar{p}\bar{q}) \otimes I\chi$ , up to overall phases. We note that the following two-qubit logical errors leave the logical Bell states unchanged and therefore do not influence the experimental outcome:  $\bar{O}\chi = \chi$ , for  $\bar{O} \in \{II, \bar{X}\bar{X}, \bar{Y}\bar{Y}, \bar{Z}\bar{Z}\}$ .

We thus proceed as follows: instead of applying  $U_{\text{enc},\chi}^\dagger$  at  $t = \tau$  we deliberately unencode into a different logical Bell state  $\chi'$  [see Fig. 1(a)]. In this manner, each of the even-weight bitstrings gives us a measure of the occurrence of one of the logical errors  $\mathcal{E}_{\bar{O},\tau}$ , where  $\bar{O}$  is determined by the unencoding we choose [see Fig. 1(b,c)]. Using this methodology, we can detect the occurrence of different types of logical errors and quantify the associated error probability.

In more detail, each row in Fig. 1(c) shows a different initially encoded logical Bell state. Each column corresponds to one of the unencoding circuits. In each unencoding scenario, the different bitstrings indicate that the state being unencoded is either the initial logical Bell state or some other logical Bell state. For example, consider the case where we originally encode  $|\Phi_+\rangle$  and then unencode into  $|\Phi_-\rangle$  at  $t = \tau$ . If the state being unencoded is indeed  $|\Phi_-\rangle = \bar{Z}_i |\Phi_+\rangle$ , then the 0000 bitstring signals that a  $Z$  logical error ( $\bar{Z}_i$ ) has occurred. However, if no error has occurred, then due to unencoding into  $|\Phi_-\rangle$ , the result should be the 0110 bitstring. Generalizing, it is possible to detect the different logical errors using the experiments indicated in Fig. 1(c).

However, we specifically choose the unencoding so that it is always the 0000 bitstring that corresponds to the occurrence of the logical error operator in which we are interested. We make this choice since  $|0000\rangle$  is the ground state of the system and therefore is robust against relaxation errors. This strongly increases the likelihood that the detected errors are purely logical and are unaffected by thermal relaxation. Note that due to this choice, the bitstring that corresponds to the fidelity  $\mathcal{F}_\chi(t)$  of the prepared initial state varies [black color-coded bitstrings in Fig. 1(c)].

From here on, we use the notation  $\chi \mapsto \chi'$  to denote the procedure of preparing the encoded logical Bell state  $\chi$  and unencoding into  $\chi'$ , i.e., of using the encoding unitary  $U_{\text{enc},\chi}$  and the unencoding unitary  $U_{\text{enc},\chi'}^\dagger$ .

We note that logical state tomography is an alternative method for learning about the performance of LDD or NDD, and in particular, for certifying the entanglement of logical Bell states. However, the  $[[4, 2, 2]]$  code imposes some additional challenges in performing logical tomography, which precluded its use in our work; see [Methods](#) for details.

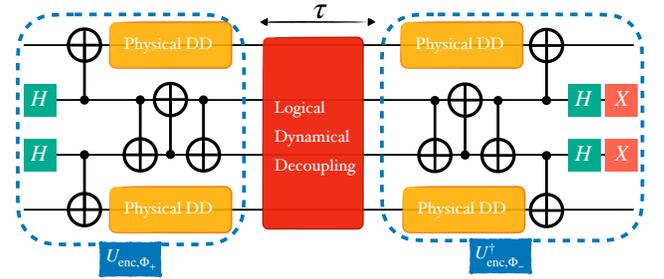


Figure 2. **Circuit schematics.** Encoding  $|\Phi_+\rangle$  and unencoding  $|\Phi_-\rangle$  with physical and logical DD. Physical XY4 dynamical decoupling sequences (yellow boxes) are inserted into the idle gaps of the circuit. The code subspace is protected with NDD sequences. This is the DD-protected version of the circuit shown in Fig. 1(a).

**Physical dynamical decoupling improves logical Bell state fidelity.** We first show that we can substantially improve the logical Bell fidelity by padding the idle gaps of the encoding circuits with physical (as opposed to logical or normalizer) DD sequences. An idle gap is a temporal circuit segment during which no gates are applied. Such gaps occur, e.g., when a pair of qubits is involved in a two-qubit gate that takes much longer than a single-qubit gate simultaneously being applied to another qubit; the latter is then idle after the completion of the single-qubit gate, while awaiting the completion of the two-qubit gate.

As shown in Fig. 2, we inserted various DD sequences into the idle gaps of Bell state circuits. This includes XY4 [27], universally robust sequences UR<sub>n</sub> [41] for  $n = 6, 8, 10, 18$ , and RGAs<sub>a</sub> [40].

Since the UR<sub>n</sub> sequences rely heavily on the use of both positive and negative rotations, it is important to clarify that henceforth, we use the notation  $X \equiv R_x(\pi) = \exp(-i\pi\sigma^x/2)$ ,  $\check{X} \equiv R_x(-\pi) = \exp(i\pi\sigma^x/2)$ , and like-

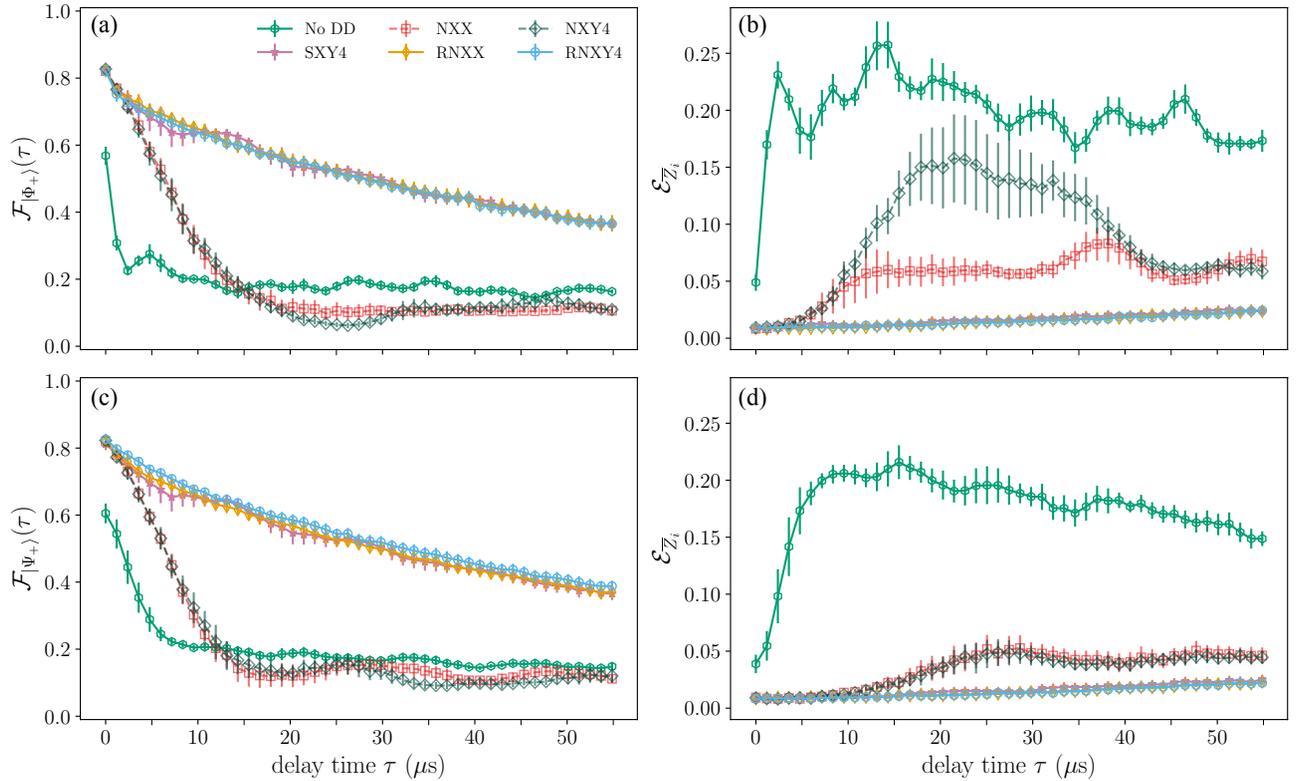


Figure 3. **Performance of NDD (dataset 1).** The experiments encode the logical Bell states  $|\Phi_{+}\rangle$  in (a) and (b), and  $|\Psi_{+}\rangle$  in (c) and (d). We then unencode in the logical Bell state related through  $\bar{Z}_i$  on either of the logical qubits; i.e.,  $|\Phi_{+}\rangle \mapsto |\Phi_{-}\rangle$  in (a) and (b), and  $|\Psi_{+}\rangle \mapsto |\Psi_{-}\rangle$  in (c) and (d). In this setting, the frequency of 0110 outcomes [(a) and (c)] is a measure of the fidelity of the originally encoded state, while 0000 corresponds to the detection of a  $\bar{Z}_i$  error [(b) and (d)]. Without any DD, the fidelity [(a) and (c)] is low and the probability of logical errors [(b) and (d)] is high. Standard NDD sequences (NXX and NXY4) improve the fidelity and logical error probability at short times ( $\lesssim 15\mu\text{s}$ ) but their performance declines at longer times: they exhibit coherent errors as seen in the oscillations of the corresponding curves. Robust sequences (RNXX and RNXY4) are the top performers; their ability to suppress the  $\bar{Z}_i$  error is particularly noteworthy. Error bars are  $1\sigma$  standard deviation after bootstrapping the data.

wise for  $Y$ , to denote physical DD pulses. That is, unlike the case described for  $\tilde{\mathcal{G}} = \tilde{\mathcal{P}}_n$ , the symbols  $X$  and  $Y$  no longer denote the Pauli  $\sigma^x$  and  $\sigma^y$  operators. The reason this matters is that physical DD pulses suffer from control errors (both axis-angle and angle-magnitude errors), as well as errors due to the presence of the system-bath interaction while the pulse is on. In other words, the notation  $R_x(\pi)$ , etc., hides the fact that in reality the exact rotation axis deviates from  $x$  and the exact rotation angle deviates from  $\pi$ . When these deviations are accounted for, DD sequences that are formally identical due to phase cancellation, such as  $XX$  and  $X\check{X}$ , result in different operations at the physical level. For more details, see [Methods](#).

We performed the experiments (during the week of August 12<sup>th</sup> 2024) on fourteen different sets of qubits of `ibm_kyiv` and report the average fidelity values in [Table II](#). In these experiments, we encoded and unencoded the same state, i.e., used  $U_{\text{enc},\chi}$  along with  $U_{\text{enc},\chi}^{\dagger}$ . We see that using DD, the encoding fidelity improves by  $\sim 20\%$  for all four logical Bell states, with XY4 and RGA8<sub>a</sub>

being the top performers (we attribute the lower performance of the longer UR<sub>n</sub> sequences to pulse interference effects [59]).

In all, these results demonstrate that logical Bell state preservation can benefit significantly from physical DD. Having established the utility of physical DD, we proceed to combine it with NDD in the next section.

**Logical error suppression and detection by NDD.** For all our experimental results, “No DD” refers to  $[[4, 2, 2]]$  encoding without any physical or logical DD. In all other experiments, we use physical XY4 to pad all the encoding and unencoding idle gaps, and combine them with various flavors of NDD. This choice allows us to clearly assess the improvements introduced by DD, compared to relying solely on the code’s error detection capabilities.

**Logical Z errors.** We start by gradually increasing the time delay  $\tau$  (up to  $55\mu\text{s}$ ) between the encoding and unencoding without DD. This situation is relevant in the context of QEC experiments. For example, one could prepare an encoded qubit and then leave it to idle

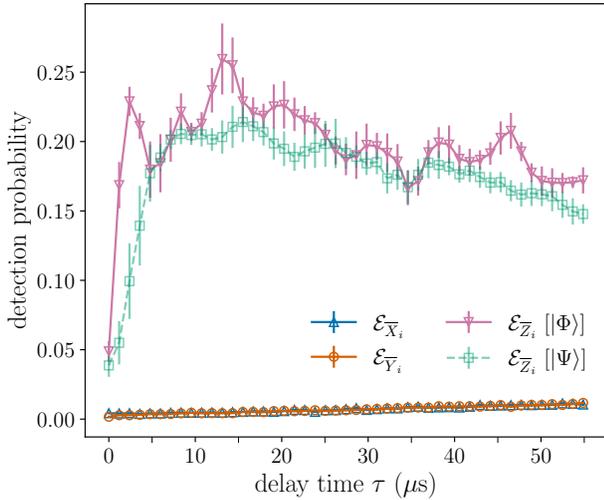


Figure 4. **Logical error detection.** Probability of the 0000 bitstring, for logical  $X$ ,  $Y$ , and  $Z$ -type errors (dataset 1). The lower probability of logical  $X$  and  $Y$ -type errors signals that the main source of logical errors is  $ZZ$  crosstalk. These experiments are performed without DD.

while other logical operations are applied to other encoded qubits [14].

We perform the  $\bar{Z}_i$  error detection using the logical Bell states  $|\Psi_+\rangle$  and  $|\Phi_+\rangle$ , and display the results in Fig. 3. As can be seen in Fig. 3(a) and (c), without NDD the free evolution fidelity (denoted as No DD) decays rapidly and exhibits  $ZZ$  crosstalk-induced oscillations. Figure 3(b) and (d) show that logical  $Z$  errors accumulate over time.

We next use two types of NDD sequences to suppress this effect: Normalizer  $XX$  (NXX) and Normalizer  $XY4$  (NXY4). Physical-level schematics of these NDD sequences are shown in the boxed four-pulse sequences of Fig. 1(d) and (e), respectively (disregarding the tilde notation). Their logical-level counterparts are shown at the bottom of Fig. 1. We generate both NXX and NXY4 using the native logical operations of the  $[[4, 2, 2]]$  code. For example, in the upper part of Fig. 1(d), reading the first (second) column from top to bottom yields  $XIXI = \bar{X}\bar{X}$  ( $\bar{X}'\bar{X}' = IXIX = S_1\bar{X}\bar{X}$ ;  $S_1 = XXXX$ ), which is the  $\bar{X}\bar{X}$  ( $\bar{X}'\bar{X}'$ ) column at the bottom of Fig. 1(d). Reading the pulse sequences from left to right, the staggering (appearance of delays as indicated by the identity operations) is deliberately introduced to suppress crosstalk at the physical level [43]; see Methods for more details.

The corresponding results are denoted as NXX (red) and NXY4 (dark green) in Fig. 3. NXX is a single-axis non-universal sequence, while NXY4 is “nearly” universal: it suppresses all logical errors except its own generating set  $\{XIXI, IXIX, YIYI, IYIY\}$ . At short times ( $\lesssim 15\mu\text{s}$ ), these results are better than those without DD because some errors (including  $ZZ$  crosstalk) are suppressed by both NXX and NXY4. However, at longer times, the benefit is lost and, moreover, small oscillations appear that indicate the presence of coherent errors. To

overcome this, we create robust versions of NXY4 and NXX by ensuring that all physical qubits in the code undergo physical DD sequences robust to pulse errors. Specifically, we use the universally robust (UR) sequence family [41], and ensure that each physical qubit undergoes a  $UR_4$  sequence, i.e.,  $X\tilde{X}\tilde{X}X$  or  $Y\tilde{Y}\tilde{Y}Y$ , where a tilde denotes an  $X$  or  $Y$  rotation by  $-\pi$  instead of  $\pi$ . These robust versions, which we call RNXX and RNXY4, are the full sequences shown in Fig. 1(d) and (e). The performance of RNXX (yellow) and RNXY4 (blue), as seen in Fig. 3, exhibits a significant improvement. Notably, the logical Bell state fidelities decay more slowly and without oscillations [panels (a) and (c)], and the logical  $Z$  errors are strongly suppressed [panels (b) and (d)].

In addition, we apply the physical staggered  $XY4$  (SXY4) sequence. This corresponds to applying a single  $XY4$  sequence to each physical qubit but in a staggered manner to reduce crosstalk [43]. This sequence, shown in Fig. 1(f), also corresponds to a non-universal purely logical- $Z$ -error suppressing, NDD sequence for the  $[[4, 2, 2]]$  code [bottom of panel (f)]. Interestingly, Fig. 3 shows that SXY4 performs on par with RNXX and RNXY4. To explain this, note that staggered physical DD on the  $[[4, 2, 2]]$  code implements a logical sequence, therefore, on the  $[[4, 2, 2]]$  code, per-qubit SXY4 and logical NDD are not independent sequences; fair disambiguation of ‘logical vs physical’ should be assessed on other codes where this lack of sequence independence does not occur. In addition, this on-par-performance finding signals that the dominant logical errors are of  $Z$ -type. We confirm this in the following.

**Logical  $X$  and  $Y$  errors.** So far, we have only discussed the detection and suppression of  $Z$ -type logical errors. In order to detect  $X$  and  $Y$ -type logical errors, we proceed in analogy to the  $\bar{Z}$  error detection procedure, but with a different unencoding step. Namely, we start by encoding the logical  $|\Phi_+\rangle$  state, and then unencode in either  $|\Psi_+\rangle$  to detect  $X$ -type errors or in  $|\Psi_-\rangle$  to detect  $Y$ -type errors. The results are shown in Fig. 4 where, for comparison, we have also included the no-DD  $|\Phi_+\rangle \mapsto |\Phi_-\rangle$  and  $|\Psi_+\rangle \mapsto |\Psi_-\rangle$  results shown in Fig. 3(b,d), which measure  $Z$ -type logical errors. It is clear from Fig. 4 that while unsuppressed logical errors  $Z$  accumulate rapidly, logical errors of  $X$  and  $Y$ -type grow much more slowly. This confirms that  $ZZ$  crosstalk is the main source of logical errors.

**NDD with postselection.** Having implemented NDD, we can further improve the results by using the error detection capability of the  $[[4, 2, 2]]$  code, which allows us to perform postselection. We do so by discarding any measurement outcome outside the logical basis (which would be the result of a physical error); i.e., we only keep measurement results corresponding to the bitstrings  $\{0000, 0110, 1001, 1111\}$ . This allows us to post-select on no detected syndrome. Note that a multi-round schedule that aborts on first fault detection selects the same class of “no-event” trajectories as our end-of-interval check, while adding measurement/ancilla overhead. The per-

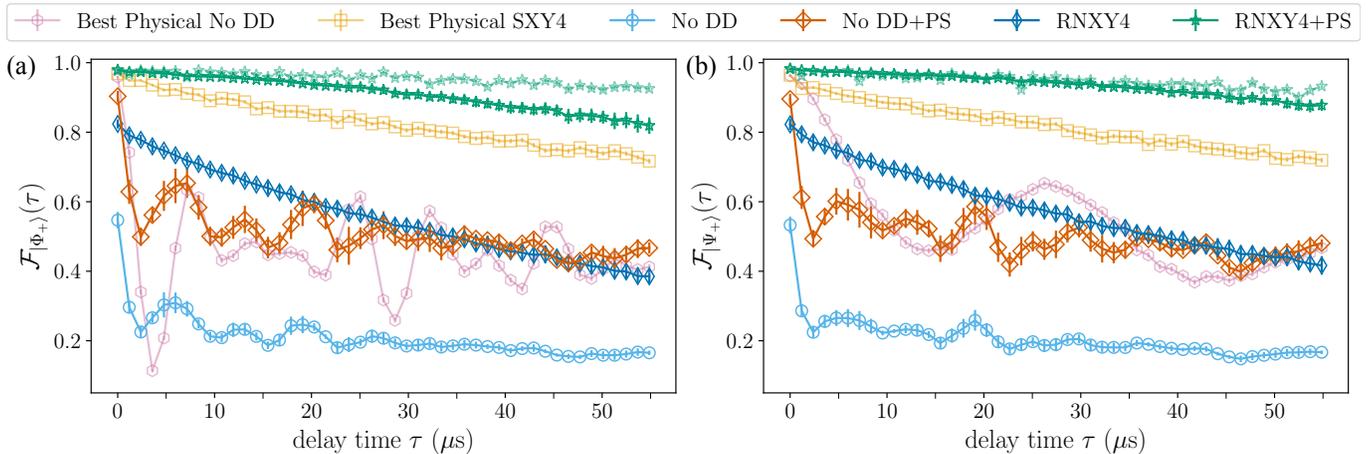


Figure 5. **Main results reporting the Fidelity of logical Bell states (dataset 2).** (a)  $|\Phi_+\rangle$  (using  $|\Phi_+\rangle \mapsto |\Phi_-\rangle$ ) and (b)  $|\Psi_+\rangle$  (using  $|\Psi_+\rangle \mapsto |\Psi_-\rangle$ ). In both cases, without DD, the fidelity drops to 20% in  $\approx 10\mu\text{s}$ . Note that for `ibm_kyiv`, median  $T_1 = 279.92\ \mu\text{s}$ , median  $T_2 = 111.926\ \mu\text{s}$ . Using error detection and applying postselection, the fidelity improves to an average of  $\approx 40\%$  over the entire  $55\mu\text{s}$  (No DD+PS). Using NDD alone is roughly equivalent to No DD+PS, but removes the crosstalk oscillations (RNXY4). When we combine NDD with postselection, the fidelity increases substantially and remains above 82% [in (a)] and 87% [in (b)] over a  $55\mu\text{s}$  period (RNXY4+PS). The light green dashed lines in each panel represent the results obtained from running the respective protocol on the best-performing set of physical qubits (using qubits 58, 59, 60, 61). In this case, RNXY4+PS yields  $\mathcal{F} \approx 92\%$  after  $55\mu\text{s}$  (for both cases), significantly higher than the fidelity using postselection after error detection alone. Also shown is the best physical Bell state prepared on the same set of qubits, which, without DD (Best Physical No DD) performs comparably to No DD+PS, though with much larger crosstalk-induced fidelity oscillations. Applying staggered XY4 (Best Physical SXY4) removes the crosstalk and improves the best physical Bell state fidelity beyond RNXY4’s (decaying to  $\mathcal{F} \approx 71\%$  after  $55\mu\text{s}$ ), but remains well below the performance of RNXY4+PS.

formance differences we report therefore isolate the effect of NDD-QED versus one-shot QED under identical post-selection and timing constraints (see [Methods](#)). Empirically, NDD both raises the fidelity (Fig. 5) and lowers the discard rate associated with detected physical faults (Fig. 6). Combining the error-detection capability of the code and NDD leads to our central experimental result, the demonstration of high-fidelity entangled logical qubits, shown in Fig. 5.

In Fig. 5(a) and (b) we show the fidelities of  $|\Phi_+\rangle \mapsto |\Phi_-\rangle$  and  $|\Psi_+\rangle \mapsto |\Psi_-\rangle$  for dataset 2. Without DD, the fidelity decays rapidly while exhibiting crosstalk oscillations, as seen for dataset 1 in Fig. 3. The fidelity improves once we perform postselection as described above. This leaves us with bitstrings corresponding to logical states, but logical errors still reduce the fidelity. Using NDD in the form of the RNXY4 sequence – which suppresses both logical errors and physical errors – followed by postselection, we achieve fidelities  $> 82\%$  for  $|\Phi_+\rangle$  and  $> 87\%$  for  $|\Psi_+\rangle$  over a  $55\mu\text{s}$  period. The average fidelities over the same period are 91.12% for  $|\Phi_+\rangle$  and 93.66 % for  $|\Psi_+\rangle$ . For the particular set of qubits numbered  $\{58, 59, 60, 61\}$  (light dashed lines), we find that the combination of NDD and postselection yields average fidelities of  $\approx 95.44\%$  for  $|\Phi_+\rangle$  and  $\approx 94.78\%$  for  $|\Psi_+\rangle$ , which is significantly higher than when we use only the error detection capability of the code:  $\approx 44.81\%$  for  $|\Phi_+\rangle$  and  $\approx 48.47\%$  for  $|\Psi_+\rangle$ .

Preserving an encoded Bell pair at high fidelity over algorithmic timescales is an operational resource for

teleportation-based subroutines and for linking logical modules; our NDD-protected preservation directly targets this use-case.

#### Beyond breakeven and the state-of-the-art.

Fig. 5 also includes results for physical (unencoded) Bell states. Here we show only the best Bell pair among all pairs we tested, both without DD and with staggered (crosstalk-robust [43]) XY4. The former (‘Best Physical No DD’) exhibits strong crosstalk-induced oscillations with an overall fidelity comparable to that of the mean logical encoded Bell pairs case without DD (No DD+PS). Adding SXY4 significantly improves the fidelity and outperforms even the mean fidelity of logical Bell pairs with NDD (RNXY4). This shows that NDD by itself is not better than working with physical qubits and a crosstalk-robust DD sequence. However, physical Bell pairs with SXY4 are significantly worse than RNXY4+PS, i.e., the case of NDD with postselection on the results of the  $[[4, 2, 2]]$  code. This constitutes clear evidence of beyond-breakeven performance for our QED-NDD strategy.

Overall, it is clear that the combination of NDD and postselection significantly boosts the fidelity of logical Bell states. Moreover, our results improve upon the current state-of-the-art using superconducting transmon qubits. E.g., Ref. [63] used distance  $d = 2$  surface codes to encode the four logical Bell states, with a peak encoding fidelity of 79.5%. In contrast, we find an average postselected encoding fidelity of 98.05%. The averaging is over the 9 sets of qubits and over the two logical Bell

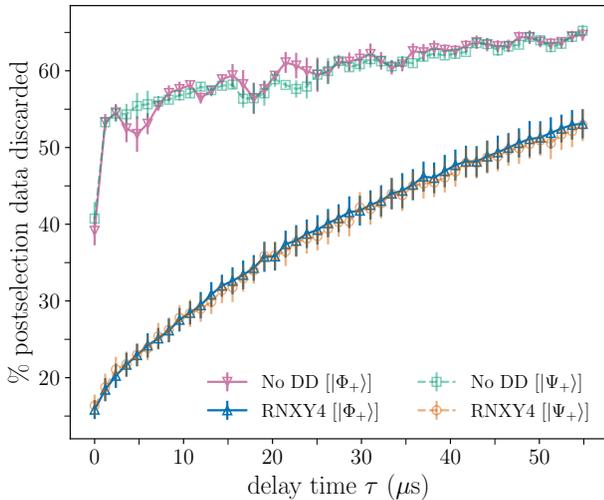


Figure 6. **Effect of NDD on postselection.** The percentage of data discarded per circuit as a function of the delay time between encoding and unencoding (dataset 2). While NDD suppresses logical errors, it also reduces certain physical errors, as evidenced by the lower data discard rate when using NDD (i.e., fewer physical errors are detected).

states we prepare.

Ref. [64] used the heavy-hex surface code with variable distance  $d$  and reported a peak postselected fidelity of 93.7% after the first stabilizer round for  $d = 2$ , declining to  $\approx 30\%$  after five rounds, which corresponds to  $\approx 27\mu\text{s}$  on `ibm_torino`. In contrast, we find an average peak postselected encoding fidelity of 98.05% that declines to 84.87% after  $55\mu\text{s}$ , and a peak postselected fidelity of 98.00% (also averaged over the two logical Bell states we prepare) for the best set of qubits, that declines to 92.89% after  $55\mu\text{s}$ , while also lowering the discard rate under post-selection.

Ref. [45] benchmarked beyond-breakeven preparation of an encoded magic state using error-suppressed circuits with either post-selection or real-time feed-forward; fidelity was evaluated immediately after preparation (logical infidelity  $\approx 1.23 \times 10^{-2}$  with post-selection;  $\approx 1.87 \times 10^{-2}$  with feed-forward), and the main idling arose from control-flow latencies of order hundreds of nanoseconds during feed-forward. Informally, their result answers ‘How good is the state right after it is prepared?’, whereas our study answers ‘How long can the state be kept good?’

**Physical error suppression and detection by NDD.** As explained in the discussion following Theorem 1 and with illustration of NDD with the  $[[4, 2, 2]]$  code, the NDD sequences suppress not only logical errors but physical errors as well. To see this, consider, e.g., the RNXX sequence [Fig. 1(d)]. As described in [Methods](#), reading the pulse sequence vertically, each time step of this sequence operates in the logical subspace as  $\overline{XX}$ . Simultaneously, reading the sequence horizontally over a complete round of NDD, all four physical qubits undergo the physical XX sequence (i.e.,  $X - \tau - X - \tau$ ), which sup-

presses the set of physical errors  $\{Y_i, Z_i\}_{i=1}^4$ . Similarly, for the RNXY4 sequence [Fig. 1(e)], each physical qubit undergoes  $\text{UR}_4$ , which robustly [41] suppresses the sets of physical errors  $\{Y_i, Z_i\}_{i=1,3}$  (due to the  $X\check{X}\check{X}X$  sequence) and  $\{X_i, Z_i\}_{i=2,4}$  (due to the  $Y\check{Y}\check{Y}Y$  sequence).

One way to gauge the impact of this suppression of physical errors is shown in Fig. 6, which displays the percentage of discarded data per circuit as a result of post-selection over  $55\mu\text{s}$  (out of 4000 shots), comparing No DD to RNXY4 for the two logical Bell states  $|\Phi_+\rangle$  and  $|\Psi_+\rangle$ . The results are obtained through bootstrapping across the ten sets of four qubits used in these experiments. Significantly less data is discarded with RNXY4 than without DD. Since discarded data corresponds to the detection of physical errors, this means that NDD not only enhances the fidelity of logical Bell states by suppressing logical errors, but also reduces the occurrence of physical errors.

We can go further and use Algorithmic Error Tomography [46] to identify specific physical error types. For example, the 0100 and 0010 bitstrings correspond, respectively, to physical  $Z_2, Z_4$  and  $Z_1, Z_3$ , errors in the  $|\Phi_+\rangle \mapsto |\Phi_-\rangle$  logical Bell state experiment. In Fig. 7 we show the corresponding relative bitstring counts, which are measures of detecting either of these errors. Additionally, we show the 0110 and 0000 relative bitstring counts, which correspond to the logical fidelity and the logical  $Z$  error, respectively. Without DD (dashed), we observe that the logical fidelity is low, and both logical and physical errors increase and oscillate. With DD (solid), the RNXY4 sequence strongly suppresses both types of errors.

These results demonstrate that, as claimed, NDD sequences suppress both physical and logical errors.

## DISCUSSION

The operation of QEC and QED codes is adversely affected by the occurrence of logical errors that the code cannot detect or correct. Here, we have introduced an efficient method for combining QEC or QED with dynamical decoupling, implemented in terms of the normalizer elements of the code. Our work is the first experimental demonstration of unequivocal suppression of the logical errors of a given code using this method. We designed our NDD sequences to simultaneously perform logical error suppression and to be robust DD sequences at the physical level, resistant to both control errors and crosstalk. As an added benefit, these sequences suppress many physical errors as well. Our results using the  $[[4, 2, 2]]$  code and IBM transmon qubits, featured in Fig. 5, demonstrate a beyond-breakeven fidelity of entangled logical qubits. The fidelities we report are the highest to date for entangled logical qubits using superconducting qubits.

Our findings address a need along the path toward fault-tolerant quantum computation: keeping codes rela-

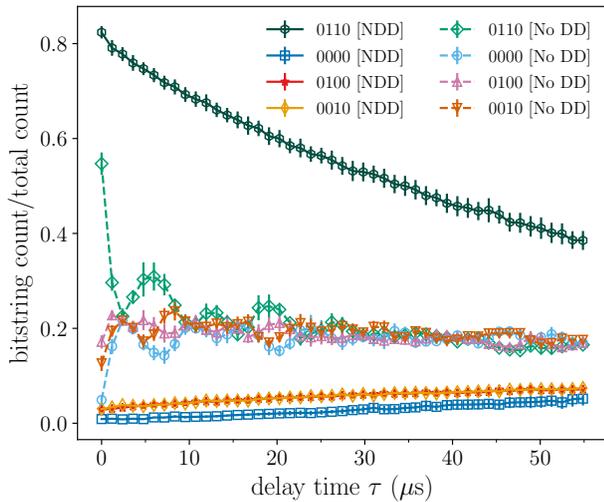


Figure 7. **Suppressing logical and physical errors.** The relative count of the  $\{0000, 0110, 0100, 0010\}$  bitstrings in the  $|\Phi_+\rangle \mapsto |\Phi_-\rangle$  experiments (dataset 2). The two even-weight bitstrings correspond to the logical fidelity (0110) and logical  $Z$  error (0000), and the two odd-weight bitstrings correspond to the detection of physical  $\{Z_i\}_{i=1}^4$  errors. Dashed (solid) lines correspond to no DD (RNXY4). The NDD sequence, in addition to suppressing the logical errors and improving the logical fidelity, also strongly suppresses the occurrence of these physical errors.

tively small and nimble while still effectively handling logical errors. Because an NDD sequence of length  $K = 2^r$  depends only on the chosen generator set (Definition 5; Table I), one can target dominant correlated errors with constant-length sequences independent of  $n$  or  $d$ . Scheduling normalizer pulses in idle gaps preserves cycle timing and reduces both logical-error accumulation (Figs. 3-5) and detected-fault rates (Fig. 6), thus decreasing decoder and restart overhead.

The sequence performance cannot be easily predicted based on standard reported metrics such as average  $T_1$ ,  $T_2$ , single- and two-qubit gate fidelities. As such, future research should aim to optimize NDD sequences tailored to specific codes and integrate QEC-NDD into quantum algorithms. Another interesting future direction is the optimization of QEC-NDD for cross-talk-free transmon devices with tunable couplers; we present preliminary results in [Methods](#).

## METHODS

**Quantum codes and logical errors.** A general  $[[n, k, d]]$  code encodes  $k$  logical qubits into  $n$  physical qubits with distance  $d$ . The weight  $w$  of an error is the number of physical qubits it affects simultaneously. An  $[[n, k, d]]$  stabilizer code corrects every error whose weight satisfies  $w \leq t = \lfloor \frac{d-1}{2} \rfloor$  and detects every error whose weight  $w \leq d - 1$ . A  $d = 2$  code is a pure quantum error detection (QED) code, and a distance- $d$  QEC code

can always be used as a QED code for errors of weight  $< d$ . The code distance is the minimum number of physical qubits that must experience an error to cause an undetectable logical error, i.e., an error forming a logical operation inside the code space. These logical errors can either be inherently present or result from the accumulation of lower-weight errors over time. For example, in the context of superconducting qubits with fixed-frequency couplers, a prevalent challenge is  $ZZ$  crosstalk [65], which inherently introduces weight-two errors that can present as logical errors for distance-2 codes.

An  $[[n, k, d]]$  stabilizer code  $\mathcal{C}$  is defined as the  $+1$  eigenspace of a stabilizer group  $\mathcal{S}$  of order  $2^{n-k}$  (with commuting generators  $\{S_j\}_{j=1}^{n-k}$ ). A stabilizer group is any subgroup of  $\mathcal{P}_n$  that excludes  $-I$  (which implies that it is Abelian). One can factor the  $n$ -qubit Hilbert space into  $k$  logical qubits with an associated group of canonical logical operators  $\mathcal{L} = \langle i, \bar{X}_j, \bar{Z}_j \rangle_{j=1}^k \subset \mathcal{P}_n$  (we use  $\langle \dots \rangle$  to denote a generating set;  $|\mathcal{L}| = 4^{k+1}$ ) and  $n - k$  syndrome qubits (which can be used to detect errors). Specifically, syndrome qubits can be used to detect Pauli group terms that anticommute with at least one of the  $n - k$  stabilizer generators [66]. Since  $\mathcal{S}$  is commutative, operators in  $\mathcal{S}$  can be simultaneously diagonalized. This simultaneous diagonalization partitions the Hilbert space into an orthogonal sum of  $2^{n-k}$  subspaces (known as syndrome spaces), each of dimension  $2^k$ , corresponding to  $2^{n-k}$  choices of  $\pm 1$  eigenvalues of  $S_j$  for  $j = 1, \dots, n - k$ , i.e., different values of the syndrome.  $\mathcal{C}$  is associated with the trivial (no error, corresponding to  $+1$  eigenvalue of all  $S_j$ ) syndrome, and two logical operators act identically on  $\mathcal{C}$  if they only differ by a stabilizer element, i.e., the full group of logical operators is  $\mathcal{N}(\mathcal{S}) = \mathcal{S}\mathcal{L}$ , the normalizer of  $\mathcal{S}$  in  $\mathcal{P}_n$ .

For a code  $\mathcal{C} \subset (\mathbb{C}^2)^{\otimes n}$ , the stabilizer group  $\mathcal{S}$  is uniquely defined as

$$\mathcal{S} = \{P \in \mathcal{P}_n : \forall |\psi\rangle \in \mathcal{C}, P|\psi\rangle = |\psi\rangle\}. \quad (11)$$

The normalizer (or centralizer) of  $\mathcal{S}$  in  $\mathcal{P}_n$  is the group  $\mathcal{N}(\mathcal{S})$  of Pauli operators commuting with all elements of  $\mathcal{S}$ . The normalizer  $\mathcal{N}(\mathcal{S}) = \mathcal{N}_{\mathcal{G}}(\mathcal{S})$  of  $\mathcal{S} \subseteq \mathcal{G}$  is defined as  $\{g \in \mathcal{G} \text{ s.t. } g\mathcal{S}g^{-1} = \mathcal{S}\}$ , and the centralizer  $\mathcal{Z}(\mathcal{S}) = \mathcal{Z}_{\mathcal{G}}(\mathcal{S})$  of  $\mathcal{S} \subseteq \mathcal{G}$  is defined as  $\{g \in \mathcal{G} \text{ s.t. } \forall s \in \mathcal{S}, gs = sg\}$ . These two notions coincide in the case of stabilizer groups. The normalizer is also uniquely defined. However, the group of canonical logical operators is not unique: one can choose arbitrary stabilizers and multiply the canonical generators  $\bar{X}_j, \bar{Z}_j$  by those to obtain other choices of the group  $\mathcal{L}$ .

Another important set is  $\mathcal{D} = \mathcal{P}_n \setminus \mathcal{N}(\mathcal{S})$ , the set of detectable errors. Since every pair of elements of  $\mathcal{P}_n$  either commute or anticommute, and  $\mathcal{N}(\mathcal{S})$  contains all elements of  $\mathcal{P}_n$  that commute with  $\mathcal{S}$ , every element of  $\mathcal{D}$  must anticommute with at least one element of  $\mathcal{S}$ .

Table I reports only the number of DD pulses per cycle for each decoupling family; it intentionally does not include the overhead of stabilizer readout that accompanies QED/QEC when LDD is used in practice. To make

the end-to-end accounting explicit, consider an  $[[n, k, d]]$  stabilizer code with generators  $\{S_j\}_{j=1}^{n-k}$  of weights  $\{w_j\}$ . A standard ancilla-mediated parity check of a single generator  $S_j$  couples the ancilla once to each data qubit on which  $S_j$  acts nontrivially (after basis changes for  $X$ - or  $Y$ -type checks), and then measures the ancilla. In the simplest implementation this uses  $w_j$  two-qubit entangling gates (CNOT or CZ with one per non-identity factor in  $S_j$ ) plus one ancilla measurement; repeating over all generators, one QED/QEC round therefore incurs  $\sum_j w_j$  entangling gates and  $n - k$  measurements, up to device-specific parallelization constraints (which reduce latency but not this gate-count budget), and ignoring the comparatively cheap single-qubit basis-change rotations. By contrast, an NDD cycle uses  $K = 2^r$  pulses determined solely by the  $r$  chosen normalizer generators (Definition 5), requires no ancillas or measurements, and can be scheduled in idle windows between algorithmic gates. For our  $[[4, 2, 2]]$  code with  $S = \langle XXXX, ZZZZ \rangle$  (both weight-4), a naive single QED round thus requires  $w_1 + w_2 = 4 + 4 = 8$  entangling operations and two measurements, whereas our RNXY4 uses  $r = 2$  ( $K = 4$ ) logical pulses per cycle and simultaneously embeds universally-robust (UR) physical DD. Hence the observed suppression of logical errors (Figs. 3-5) and the lower discard rate under postselection (Fig. 6). This separation of costs clarifies how NDD can yield a lower end-to-end overhead.

**$[[4, 2, 2]]$  code.** The  $[[4, 2, 2]]$  code is an error detection code that encodes  $k = 2$  logical qubits into  $n = 4$  physical qubits [67]. The stabilizer group is  $S = \{I, XXXX, YYYY, ZZZZ\}$ . Defining the logical states as

$$\begin{aligned} |\overline{00}\rangle &= \frac{|0000\rangle + |1111\rangle}{\sqrt{2}}, & |\overline{10}\rangle &= \frac{|0110\rangle + |1001\rangle}{\sqrt{2}}, \\ |\overline{01}\rangle &= \frac{|0011\rangle + |1100\rangle}{\sqrt{2}}, & |\overline{11}\rangle &= \frac{|1010\rangle + |0101\rangle}{\sqrt{2}}, \end{aligned} \quad (12)$$

a set of logical operators for the code can be defined such that  $\overline{X}_1 = XIIX$ , and  $\overline{X}_2 = IIXX$ , up to multiplication by a stabilizer element. Therefore, we have  $\overline{X}\overline{X} = XIXI$ . Similar definitions apply to the logical  $Z$  operators:  $\overline{Z}_1 = IIZZ$  and  $\overline{Z}_2 = ZIIZ$ , allowing us to form the full logical Pauli group. Using the same definitions, we have  $\overline{\text{CNOT}}_{12} = \text{SWAP}_{12}$  and similarly,  $\overline{\text{CNOT}}_{21} = \text{SWAP}_{23}$ , i.e., we can perform logical CNOTs by swapping the physical qubits.

Our particular choice of logical operators is motivated by the fact that  $\overline{X}\overline{X}$  and  $\overline{Y}\overline{Y}$  (i.e., the logical operators we use to implement the LDD or NDD sequences) have a natural staggering of their physical  $X$  and  $Y$  gates. In other words, in the implementation of the logical operators comprising NDD, nearest-neighbor qubits are always interleaved with an identity operation (e.g.,  $I XIX$  as opposed to  $I XXI$ ). This is critical for nearest-neighbor crosstalk cancellation [43].

To encode the logical Bell states  $|\Psi_{\pm}\rangle$  and  $|\Phi_{\pm}\rangle$  using

the two logical qubits of the  $[[4, 2, 2]]$  code, consider:

$$|\Phi_{\pm}\rangle = \frac{1}{\sqrt{2}}(|\overline{00}\rangle \pm |\overline{11}\rangle) \quad (13a)$$

$$= \frac{1}{\sqrt{2}}(|0000\rangle + |1111\rangle \pm |0101\rangle \pm |1010\rangle) \quad (13b)$$

$$= \overline{\text{CNOT}}_{21} \frac{1}{\sqrt{2}}[|0000\rangle + |1111\rangle \pm |0011\rangle \pm |1100\rangle] \quad (13c)$$

$$= \overline{\text{CNOT}}_{21} \frac{1}{\sqrt{2}}[(|00\rangle \pm |11\rangle)(|00\rangle \pm |11\rangle)]. \quad (13d)$$

Thus, we proceed by preparing two *physical* copies of the target Bell state on the four physical qubits, then apply  $\overline{\text{CNOT}}_{21}$ , which creates the intended Bell state. The  $|\Psi_{\pm}\rangle$  Bell states are prepared similarly.

**Construction of our  $[[4, 2, 2]]$  code subgroup-NDD sequence.** One of the limitations of our experimental setup is that physical  $Z$  gates are unavailable: fixed-frequency transmons utilize virtual- $Z$  gates [59, 68]. Such gates are problematic, as DD sequences rely on physical rotations of the qubit [59]. This highlights a critical aspect of using NDD: by judiciously choosing the normalizers, one can eliminate the need for  $Z$  gates in the logical operators. Here, now demonstrate that there are DD sequences satisfying the conditions of Theorem 1 which do not use any physical  $Z$  pulses in the context of the  $[[4, 2, 2]]$  code. This hardware-compatible construction, i.e., no physical  $Z$  and UR physical subsequences, is essential to making code-structured DD viable on current superconducting platforms.

To construct such a sequence, we first choose generators of  $\mathcal{L}$  that do not involve physical  $Z$ 's. A simple example is  $\mathcal{L} = \langle i, XIIX, IIXX, IYY, YIYY \rangle$ . One could follow the construction below with this choice. However, in order to improve the ability of the DD sequence to suppress  $ZZ$  crosstalk, we make a different choice. First, multiply the original generators of  $\mathcal{L}$  (that is,  $XIIX, IIXX, IIZZ, ZIIZ$ ) by the stabilizers  $XXXX, XXXX, XXXX$ , and  $ZZZZ$ , respectively, to obtain  $\mathcal{L}' = \langle i, IXXI, XXII, XXY, IZZI \rangle$ . Then, after some group operations, we can write the same group using a different set of generators containing no  $Z$ 's:  $\mathcal{L}' = \langle i, XXII, IYY, IYIY, XIXI \rangle$ . The corresponding NDD sequence consisting of two repetitions of  $XIXI, IYIY, XIXI, IYY, XIXI, IYIY, XIXI$ , and  $XXII$ , which implements the decoupling group  $\mathcal{L}'/\{\pm 1, \pm i\}$ , and involves no  $Z$  gates.

This sequence was obtained as follows: denote the generators of  $\mathcal{L}$  (or  $\mathcal{L}'$ ) as  $h_1, h_2, h_3, h_4$  and choose any Gray code—a sequence of bitstrings  $a_0, \dots, a_{15}$  such that neighboring bitstrings differ in only a single digit. Then, pick  $g_j = h^{a_j}$  (for  $j = 0, \dots, 15$ ), where  $a_j$  is interpreted as a multiindex (i.e.,  $g_j = \prod_{k=1}^4 h_k^{(a_j)_k}$ ). The pulses of the DD sequence are  $g_j g_{j+1}$  for  $j = 0, \dots, 15$ .

#### One-shot vs repeated QED post-selection.

Consider an  $[[n, k, d]]$  code with stabilizer generators  $S_{j=1}^{n-k}$ . Let a detectable-fault be any event that takes

the state outside the codespace (i.e., anticommutes with at least one  $S_j$ ). A post-select-on-no-syndrome protocol keeps only trajectories in which no detectable fault occurs during the storage interval  $[0, \tau]$ .

If stabilizers are measured at rounds  $t_1, \dots, t_m$  with abort-on-fault-detection, the surviving trajectories are exactly those with zero detectable faults on  $[0, \tau]$ . With a single check at  $t = \tau$  (our protocol), the surviving trajectories are also those with zero detectable faults on  $[0, \tau]$ , except for the rare case where a sequence of two or more detectable faults returns the system to the codespace before  $t = \tau$ . Under a sparse-fault model (Markovian faults with rate  $\lambda_{\text{det}}$ ), the probability of such events is  $O((\lambda_{\text{det}}\tau)^2)$ , which is negligible on the timescales and rates relevant here. Therefore, for our preservation task, repeated checks do not change the set of surviving trajectories used to compute conditional fidelity; they only increase runtime and introduce additional control/measurement overhead. In contrast, NDD reduces both (i) the accumulation of undetectable logical errors and (ii) the incidence of detectable faults (as seen in the reduced discard fraction; Fig. 6), which is why the combination NDD plus post-selection achieves the highest conditional fidelities Fig. 5).

**Issues involving the implementation of logical state tomography using the  $[[4, 2, 2]]$  code.** A general two-qubit state requires information about 15 expectation values which requires 9 independent measurement settings (achieved by measuring each qubit in three complementary bases; e.g.,  $X$ ,  $Y$ , and  $Z$ ) [69]. This holds for logical tomography of a logical two-qubit state as well. There are a few ways in which a logical observable can be measured. First, one can use stabilizer-measurement-like circuits where the logical operator is measured by executing a circuit composed of CNOT gates targeting an ancilla qubit that is measured to learn the logical measurement outcome. This method can further be paired with a round of syndrome extraction using additional ancilla qubits to learn whether the state was in the code space to begin with. This combination of measurements allows us to invoke the code's protection while simultaneously performing a measurement. The downside of this method is the additional overhead in CNOTs and ancilla qubits needed to perform the measurement protocol. In our case, this would require a substantial overhead in SWAP gates as well since the IBM QPU's heavy-hex lattice does not pair naturally with the  $[[4, 2, 2]]$  code. Using this logical measurement method would inevitably introduce more errors and reduce the accuracy of logical state tomography.

An alternative, less costly approach for performing logical measurements is to directly measure all data qubits of the code in lieu of introducing ancillas and additional CNOTs. However, this method is incompatible with extracting information about the entire stabilizer generator set while also performing the logical measurement. As a result, we would not be able to know with certainty that the system state was in the code space at the time of mea-

surement. There is some nuance to this approach, which does allow us to learn information about stabilizers which commute qubit-wise (i.e., they share the same Pauli operator or  $I$  on the same qubit) with the logical operator being measured. For example, we can simultaneously measure  $\bar{X}_1 = XIX$ ,  $\bar{X}_2 = IIX$ ,  $\bar{X}_1\bar{X}_2 = XIXI$  and the stabilizer  $S_1 = XXXX$  since these operators all qubit-wise commute. Performing the measurement of each data qubit in the  $X$  basis gives us information about each of these logical observables, as well as information about whether or not the system was in the logical code space with respect to the stabilizer generator  $S_1$ . Unfortunately, this symmetry does not hold for the operators  $\bar{Y}_1 = XIZY$ ,  $\bar{Y}_2 = ZIXY$ , and  $S_2 = YYY$  since these operators do not all commute qubit-wise. This means that we cannot simultaneously learn information about these logical observables and information about whether or not the system was in the code space with respect to the stabilizer  $S_2$  in the same shot, reducing our total information about the system. This would lead to a less precise estimation of the logical density matrix, as we will count more experimental shots involving the system outside of the logical state space in our estimation of the logical expectation value.

For these reasons, we did not use logical state tomography in this work.

#### Creating robust logical sequences.

As discussed in the main text, we use the logical operators  $\bar{X}\bar{X} = XIXI$  and  $\bar{Y}\bar{Y} = IYIY \equiv YIYI$  (along with identity operators) to generate the LDD groups  $\mathcal{G}_{\text{LXY4}} = \{I, \bar{X}\bar{X}, \bar{Y}\bar{Y}, ZZ\}$ , and  $\mathcal{G}_{\text{LXX}} = \{I, \bar{X}\bar{X}\}$ . Cycling over the group elements creates the basis for the four-pulse sequences shown in the red boxes of Fig. 1(d,e). However, we add two additional components to enhance crosstalk cancellation and robustness.

Since the main source of logical errors is  $ZZ$  crosstalk, we modify LDD so that it cancels such errors. Thus, in the case of LXX, we insert a stabilizer every other pulse-step to consistently apply a staggered sequence to all qubits; in this manner, we move from LDD to NDD. These sequences correspond to applying logical  $X$  and  $Y$  operators to both logical qubits.

To create the robust sequences, we mirror the original sequences (hence the eight-pulse sequences) and instead of  $X$  and  $Y$ , we apply  $\check{X} = R_x(-\pi)$ ,  $\check{Y} = R_y(-\pi)$ , respectively, such that each sequence undergoes a robust pulse sequence at the physical level. We also apply staggered physical  $XY4$  to each qubit as shown in Fig. 1(f). This sequence is also inherently robust at the physical level since each qubit receives an  $XY4$  sequence, which is robust to pulse errors [41]. However, it is not a universal decoupling sequence at the logical level, as seen in the figure (i.e., it lacks the ability to decouple arbitrary single logical-qubit errors).

**Normalizer dynamical decoupling with tunable couplers.** Here we present additional experiments on `ibm_marrakesh`, which features tunable couplers [70] unlike the always-on  $ZZ$  interaction in `ibm_kyiv`. Con-

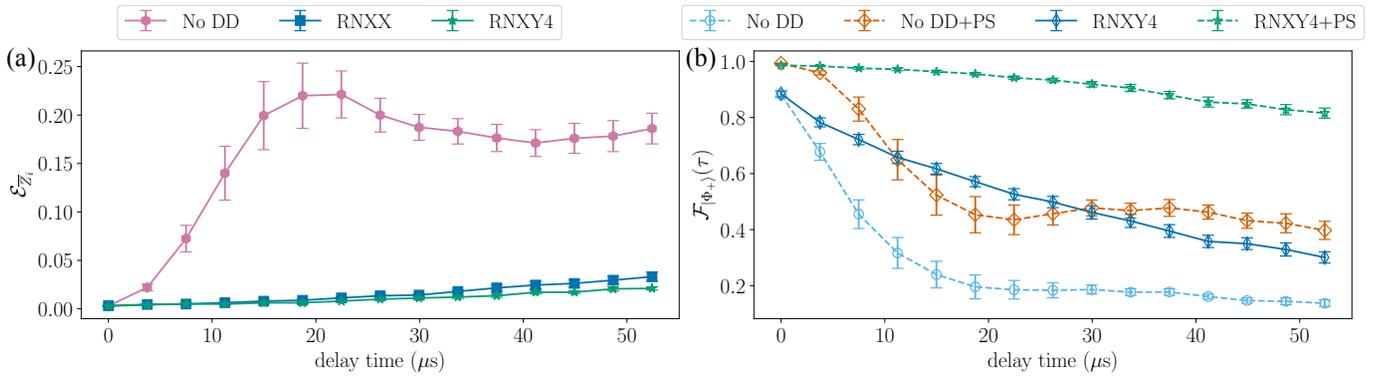


Figure 8. **NDD with tunable couplers.** An encoded  $|\Phi_+\rangle$  Bell state on `ibm_marrakesh` for a single set of qubits. (a) shows the logical errors and (b) shows the postselection results. This device features tunable couplers that significantly reduce the crosstalk. Nevertheless, accumulation of dephasing errors over time (up to  $\approx 55 \mu\text{s}$ ) leads to the appearance of logical errors. The same NDD sequence we used in the main text successfully suppresses these errors, as shown in (a). Additionally, NDD with postselection outperforms postselection alone, as shown in (b).

sequently, for this set of experiments, crosstalk is significantly reduced compared to the other set of results ( $\lesssim 5 \text{ kHz}$  vs tens of kHz). Nevertheless, a combination of dephasing and residual crosstalk accumulates, leading to logical errors. Figure 8 illustrates the performance of NDD sequences on this QPU, averaged over 9 different qubit sets. Figure 8(a) [equivalent to Fig. 3(b)] shows that without DD, logical errors remain unsuppressed. However, both NDD sequences we employ successfully suppress these errors. Figure 8(b) [equivalent to Fig. 5(a)] demonstrates that using only postselection, the fidelity averages to  $\mathcal{F} \approx 56.23\%$ , whereas combining postselection with NDD increases it to  $\mathcal{F} \approx 91.73\%$ .

#### DATA AVAILABILITY

All data are available from the corresponding authors upon request.

#### CODE AVAILABILITY

Qiskit experiments and analysis codes that support the findings of this study are available from the corresponding authors upon request.

- [1] P. W. Shor, Scheme for reducing decoherence in quantum computer memory, *Phys. Rev. A* **52**, R2493 (1995).
- [2] A. M. Steane, Error correcting codes in quantum theory, *Phys. Rev. Lett.* **77**, 793 (1996).
- [3] A. R. Calderbank and P. W. Shor, Good quantum error-correcting codes exist, *Physical Review A* **54**, 1098 (1996).
- [4] D. Aharonov and M. Ben-Or, Fault tolerant quantum computation with constant error, in *Proceedings of 29th Annual ACM Symposium on Theory of Computing (STOC)* (ACM, New York, NY, 1997) p. 176.
- [5] D. Gottesman, Theory of fault-tolerant quantum computation, *Physical Review A* **57**, 127 (1998).
- [6] E. Knill, R. Laflamme, and W. H. Zurek, Resilient quantum computation, *Science* **279**, 342 (1998).
- [7] D. Lidar and T. Brun, eds., *Quantum Error Correction* (Cambridge University Press, Cambridge, UK, 2013).
- [8] E. T. Campbell, B. M. Terhal, and C. Vuillot, Roads towards fault-tolerant universal quantum computation, *Nature* **549**, 172 EP (2017).
- [9] M. Takita, A. W. Cross, A. D. Córcoles, J. M. Chow, and J. M. Gambetta, Experimental demonstration of fault-tolerant state preparation with superconducting qubits, *Physical Review Letters* **119**, 180501 (2017).
- [10] R. Harper and S. T. Flammia, Fault-tolerant logical gates in the `ibm` quantum experience, *Physical Review Letters* **122**, 080504 (2019).
- [11] N. M. Linke, M. Gutierrez, K. A. Landsman, C. Figgatt, S. Debnath, K. R. Brown, and C. Monroe, Fault-tolerant quantum error detection, *Science Advances* **3**, e1701074 (2017).
- [12] S. Krinner, N. Lacroix, A. Remm, A. Di Paolo, E. Genois, C. Leroux, C. Hellings, S. Lazar, F. Swiadek, J. Herrmann, G. J. Norris, C. K. Andersen, M. Müller, A. Blais, C. Eichler, and A. Wallraff, Realizing repeated quantum error correction in a distance-three surface code, *Nature* **605**, 669 (2022).
- [13] Google Quantum AI and collaborators, Suppressing quantum errors by scaling a surface code logical qubit, *Nature* **614**, 676 (2023).
- [14] D. Bluvstein, S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kalinowski, D. Hangleiter, J. P. Bonilla Ataides, N. Maskara, I. Cong, X. Gao, P. Sales Rodriguez, T. Karolyshyn, G. Semeghini, M. J. Gullans, M. Greiner, V. Vuletić, and M. D. Lukin, Logical quantum processor based on

- reconfigurable atom arrays, *Nature* **626**, 58 (2023).
- [15] L. Postler, F. Butt, I. Pogorelov, C. D. Marciniak, S. Heußen, R. Blatt, P. Schindler, M. Rispler, M. Müller, and T. Monz, Demonstration of fault-tolerant steane quantum error correction, *PRX Quantum* **5**, 030326 (2024).
- [16] Google Quantum AI and collaborators, Quantum error correction below the surface code threshold, *Nature* **638**, 920 (2024).
- [17] M. Reiher, N. Wiebe, K. M. Svore, D. Wecker, and M. Troyer, Elucidating reaction mechanisms on quantum computers, *Proceedings of the National Academy of Sciences*, 201619152 (2017).
- [18] C. Gidney, *How to factor 2048 bit rsa integers with less than a million noisy qubits* (2025), [arXiv:2505.15917 \[quant-ph\]](https://arxiv.org/abs/2505.15917).
- [19] A. M. Dalzell, S. McArdle, M. Berta, P. Bienias, C.-F. Chen, A. Gilyén, C. T. Hann, M. J. Kastoryano, E. T. Khabiboulline, A. Kubica, G. Salton, S. Wang, and F. G. S. L. Brandão, *Quantum algorithms: A survey of applications and end-to-end complexities* (2023), [arXiv:2310.03011 \[quant-ph\]](https://arxiv.org/abs/2310.03011).
- [20] J. Preskill, Sufficient condition on noise correlations for scalable quantum computing, *Quant. Inf. Comput.* **13**, 181 (2013).
- [21] B. D. Clader, C. J. Trout, J. P. Barnes, K. Schultz, G. Quiroz, and P. Titum, Impact of correlations and heavy tails on quantum error correction, *Physical Review A* **103**, 052428 (2021).
- [22] E. Knill, R. Laflamme, *Concatenated Quantum Codes* (1996), [quant-ph/9608012](https://arxiv.org/abs/quant-ph/9608012).
- [23] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Topological quantum memory, *Journal of Mathematical Physics* **43**, 4452 (2002).
- [24] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Surface codes: Towards practical large-scale quantum computation, *Phys. Rev. A* **86**, 032324 (2012).
- [25] H. Bombin and M. A. Martin-Delgado, Topological quantum distillation, *Physical Review Letters* **97**, 180501 (2006).
- [26] N. P. Breuckmann and J. N. Eberhardt, Quantum low-density parity-check codes, *PRX Quantum* **2**, 040101 (2021).
- [27] L. Viola, E. Knill, and S. Lloyd, Dynamical decoupling of open quantum systems, *Phys. Rev. Lett.* **82**, 2417 (1999).
- [28] V. Tripathi, N. Goss, A. Vezvae, L. B. Nguyen, I. Siddiqi, and D. A. Lidar, Qudit dynamical decoupling on a superconducting quantum processor, *Phys. Rev. Lett.* **134**, 050601 (2025).
- [29] A. M. Souza, Process tomography of robust dynamical decoupling with superconducting qubits, *Quantum Information Processing* **20**, [10.1007/s11128-021-03176-z](https://doi.org/10.1007/s11128-021-03176-z) (2021).
- [30] A. Rahman, D. J. Egger, and C. Arenz, Learning how to dynamically decouple by optimizing rotational gates, *Physical Review Applied* **22**, 054074 (2024).
- [31] 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. Kruslich, 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, Demonstration of quantum volume 64 on a superconducting quantum computing system, *Quantum Sci. Technol.* **6**, 025020 (2021).
- [32] E. Bäumer, V. Tripathi, D. S. Wang, P. Rall, E. H. Chen, S. Majumder, A. Seif, and Z. K. Mineev, Efficient long-range entanglement using dynamic circuits, *PRX Quantum* **5**, 030339 (2024).
- [33] B. Pokharel and D. A. Lidar, Demonstration of algorithmic quantum speedup, *Phys. Rev. Lett.* **130**, 210602 (2023).
- [34] P. Singkanipa, V. Kasatkin, Z. Zhou, G. Quiroz, and D. A. Lidar, Demonstration of algorithmic quantum speedup for an abelian hidden subgroup problem, *Phys. Rev. X* **15**, 021082 (2025).
- [35] H. K. Ng, D. A. Lidar, and J. Preskill, Combining dynamical decoupling with fault-tolerant quantum computation, *Phys. Rev. A* **84**, 012305 (2011).
- [36] G. A. Paz-Silva and D. A. Lidar, Optimally combining dynamical decoupling and quantum error correction, *Scientific Reports* **3**, [10.1038/srep01530](https://doi.org/10.1038/srep01530) (2013).
- [37] A. Vezvae, C. Benito, M. Morford-Oberst, A. Bermudez, and D. A. Lidar, *Surface code scaling on heavy-hex superconducting quantum processors* (2025), [arXiv:2510.18847 \[quant-ph\]](https://arxiv.org/abs/2510.18847).
- [38] H. Goto, Y. Ho, and T. Kanao, Measurement-free fault-tolerant logical-zero-state encoding of the distance-three nine-qubit surface code in a one-dimensional qubit array, *Phys. Rev. Res.* **5**, 043137 (2023).
- [39] J.-X. Han, J. Zhang, G.-M. Xue, H. Yu, and G. Long, *Protecting logical qubits with dynamical decoupling* (2024), [arXiv:2402.05604 \[quant-ph\]](https://arxiv.org/abs/2402.05604).
- [40] G. Quiroz and D. A. Lidar, Optimized dynamical decoupling via genetic algorithms, *Phys. Rev. A* **88**, 052306 (2013).
- [41] G. T. Genov, D. Schraft, N. V. Vitanov, and T. Halfmann, Arbitrarily accurate pulse sequences for robust dynamical decoupling, *Phys. Rev. Lett.* **118**, 133202 (2017).
- [42] V. Tripathi, H. Chen, M. Khezri, K.-W. Yip, E. Levenson-Falk, and D. A. Lidar, Suppression of crosstalk in superconducting qubits using dynamical decoupling, *Phys. Rev. Appl.* **18**, 024068 (2022).
- [43] Z. Zhou, R. Sitler, Y. Oda, K. Schultz, and G. Quiroz, Quantum crosstalk robust quantum control, *Phys. Rev. Lett.* **131**, 210802 (2023).
- [44] B. W. Reichardt, A. Paetznick, D. Aasen, I. Basov, J. M. Bello-Rivas, P. Bonderson, R. Chao, W. van Dam, M. B. Hastings, R. V. Mishmash, A. Paz, M. P. da Silva, A. Sundaram, K. M. Svore, A. Vashillo, Z. Wang, M. Zanner, W. B. Cairncross, C.-A. Chen, D. Crow, H. Kim, J. M. Kindem, J. King, M. McDonald, M. A. Norcia, A. Ryou, M. Stone, L. Wadleigh, K. Barnes, P. Battaglino, T. C. Bohdanowicz, G. Booth, A. Brown, M. O. Brown, K. Cassella, R. Coxe, J. M. Epstein, M. Feldkamp, C. Griger, E. Halperin, A. Heinz, F. Hummel, M. Jaffe, A. M. W. Jones, E. Kapit, K. Kotru, J. Lauigan, M. Li, J. Marjanovic, E. Megidish, M. Meredith, R. Morshead, J. A. Muniz, S. Narayanaswami, C. Nishiguchi, T. Paule, K. A. Pawlak, K. L. Pudenz, D. R. Pérez, J. Simon, A. Smull, D. Stack, M. Urbanek, R. J. M. van de Veerdonk, Z. Vendeiro, R. T. Weverka, T. Wilkason, T.-Y. Wu, X. Xie, E. Zalgaller, X. Zhang, and B. J. Bloom, *Fault-tolerant quantum computation with a neutral atom processor* (2025), [arXiv:2411.11822 \[quant-ph\]](https://arxiv.org/abs/2411.11822).

- [45] R. S. Gupta, N. Sundaresan, T. Alexander, C. J. Wood, S. T. Merkel, M. B. Healy, M. Hillenbrand, T. Jochym-O'Connor, J. R. Wootton, T. J. Yoder, A. W. Cross, M. Takita, and B. J. Brown, Encoding a magic state with beyond break-even fidelity, *Nature* **625**, 259–263 (2024).
- [46] B. Pokharel and D. Lidar, Better-than-classical Grover search via quantum error detection and suppression, *npj Quantum Information* **10** (2024).
- [47] P. Zanardi, Symmetrizing evolutions, *Physics Letters A* **258**, 77 (1999).
- [48] A. A. Maudsley, Modified carr-purcell-meiboom-gill sequence for nmr fourier imaging applications, *Journal of Magnetic Resonance* (1969) **69**, 488 (1986).
- [49] K. Khodjasteh and D. A. Lidar, Fault-tolerant quantum dynamical decoupling, *Physical Review Letters* **95**, 180501 (2005).
- [50] K. Szczygielski and R. Alicki, Markovian theory of dynamical decoupling by periodic control, *Physical Review A* **92**, 022349 (2015).
- [51] C. Addis, F. Ciccarello, M. Cascio, G. M. Palma, and S. Maniscalco, Dynamical decoupling efficiency versus quantum non-markovianity, *New Journal of Physics* **17**, 123004 (2015).
- [52] C. Arenz, D. Burgarth, P. Facchi, and R. Hillier, Dynamical decoupling of unbounded hamiltonians, *Journal of Mathematical Physics*, *Journal of Mathematical Physics* **59**, 032203 (2018).
- [53] E. Mozgunov and D. Lidar, Completely positive master equation for arbitrary driving and small level spacing, *Quantum* **4**, 227 (2020).
- [54] K. Khodjasteh and D. A. Lidar, Quantum computing in the presence of spontaneous emission by a combined dynamical decoupling and quantum-error-correction strategy, *Physical Review A* **68**, 022322 (2003), erratum: *ibid*, *Phys. Rev. A* **72**, 029905 (2005).
- [55] P. Zanardi, D. A. Lidar, and S. Lloyd, Quantum tensor product structures are observable induced, *Physical Review Letters* **92**, 060402 (2004).
- [56] L. Viola, E. Knill, and S. Lloyd, Dynamical generation of noiseless quantum subsystems, *Phys. Rev. Lett.* **85**, 3520 (2000).
- [57] P. Zanardi, Stabilizing quantum information, *Physical Review A* **63**, 012301 (2000).
- [58] G. Quiroz and D. A. Lidar, High-fidelity adiabatic quantum computation via dynamical decoupling, *Phys. Rev. A* **86**, 042333 (2012).
- [59] A. Vezvaei, V. Tripathi, D. Kowsari, E. Levenson-Falk, and D. A. Lidar, Virtual- $z$  gates and symmetric gate compilation, *PRX Quantum* **6**, 020348 (2025).
- [60] G. S. Uhrig, Keeping a quantum bit alive by optimized  $\pi$ -pulse sequences, *Phys. Rev. Lett.* **98**, 100504 (2007).
- [61] Z.-Y. Wang and R.-B. Liu, Protection of quantum systems by nested dynamical decoupling, *Phys. Rev. A* **83**, 022306 (2011).
- [62] A. Blais, A. L. Grimsmo, S. M. Girvin, and A. Wallraff, Circuit quantum electrodynamics, *Rev. Mod. Phys.* **93**, 025005 (2021).
- [63] J. Zhang, Z.-Y. Chen, Y.-J. Wang, B.-H. Lu, H.-F. Zhang, J.-N. Li, P. Duan, Y.-C. Wu, and G.-P. Guo, Demonstrating a universal logical gate set in error-detecting surface codes on a superconducting quantum processor (2024), [arXiv:2405.09035 \[quant-ph\]](https://arxiv.org/abs/2405.09035).
- [64] B. Hetényi and J. R. Wootton, Creating entangled logical qubits in the heavy-hex lattice with topological codes, *PRX Quantum* **5**, 040334 (2024).
- [65] K. X. Wei, E. Magesan, I. Lauer, S. Srinivasan, D. F. Bogorin, S. Carnevale, G. A. Keefe, Y. Kim, D. Klaus, W. Landers, N. Sundaresan, C. Wang, E. J. Zhang, M. Steffen, O. E. Dial, D. C. McKay, and A. Kandala, Hamiltonian engineering with multicolor drives for fast entangling gates and quantum crosstalk cancellation, *Physical Review Letters* **129**, 060501 (2022).
- [66] D. Gottesman, Class of quantum error-correcting codes saturating the quantum hamming bound, *Phys. Rev. A* **54**, 1862 (1996).
- [67] L. Vaidman, L. Goldenberg, and S. Wiesner, Error prevention scheme with four particles, *Physical Review A* **54**, R1745 (1996).
- [68] D. C. McKay, C. J. Wood, S. Sheldon, J. M. Chow, and J. M. Gambetta, Efficient  $z$  gates for quantum computing, *Physical Review A* **96**, 022330 (2017).
- [69] D. F. V. James, P. G. Kwiat, W. J. Munro, and A. G. White, Measurement of qubits, *Physical Review A* **64**, 052312 (2001).
- [70] J. Stehlik, D. M. Zajac, D. L. Underwood, T. Phung, J. Blair, S. Carnevale, D. Klaus, G. A. Keefe, A. Carniol, M. Kumph, M. Steffen, and O. E. Dial, Tunable coupling architecture for fixed-frequency transmon superconducting qubits, *Phys. Rev. Lett.* **127**, 080505 (2021).

## ACKNOWLEDGMENTS

This material is based upon work supported by, or in part by, the Intelligence Advanced Research Projects Activity (IARPA), under the Entangled Logical Qubits program through Cooperative Agreement Number W911NF23-2-0216, by the U.S. Army Research Laboratory and the U.S. Army Research Office under contract/grant number W911NF2310255, and by the Defense Advanced Research Projects Agency under Agreement HR00112230006. The views, opinions and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. This research was conducted using IBM Quantum Systems provided through the University of Southern California's IBM Quantum Innovation Center. The views expressed are those of the authors and do not reflect the official policy or position of IBM or the IBM Quantum team.

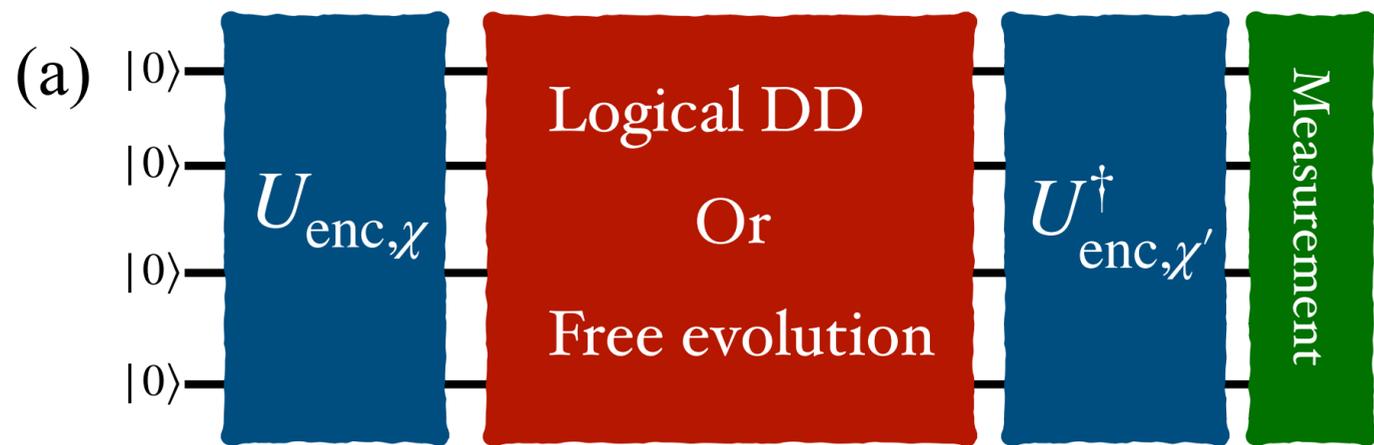
## AUTHOR CONTRIBUTIONS

A.V. and D.A.L. conceived the project. A.V. designed and performed the experiments. A.V., V.T., M.M.-O., and D.A.L. analyzed the data and results. V.T. designed the robust structure of sequences. M.M.-O., V.K., F.B., and D.A.L. designed and constructed the definitions and proofs of the theorems. All authors contributed to the writing of the manuscript.

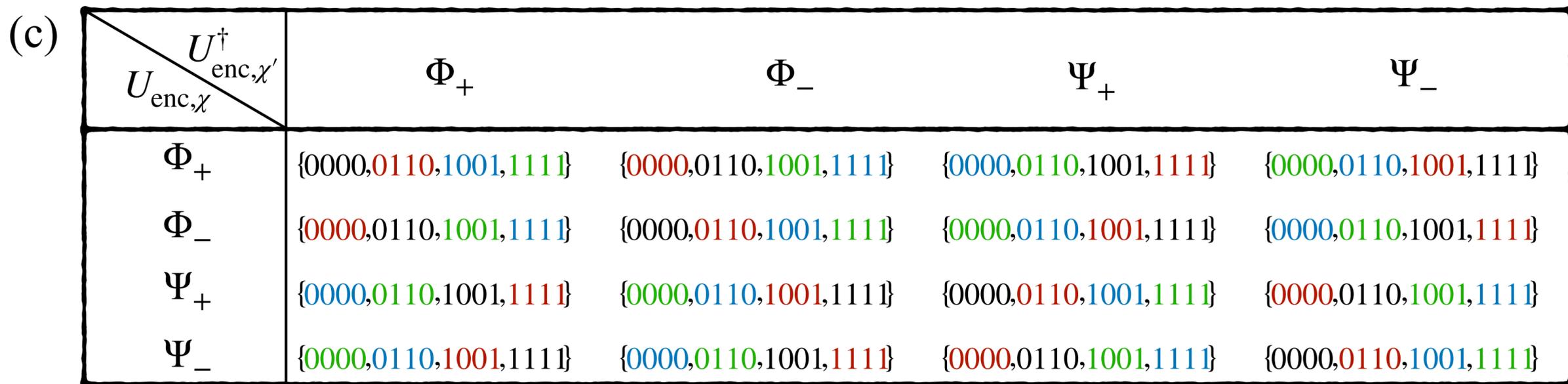
**COMPETING INTERESTS**

The authors declare no competing interests.

ARTICLE IN PRESS

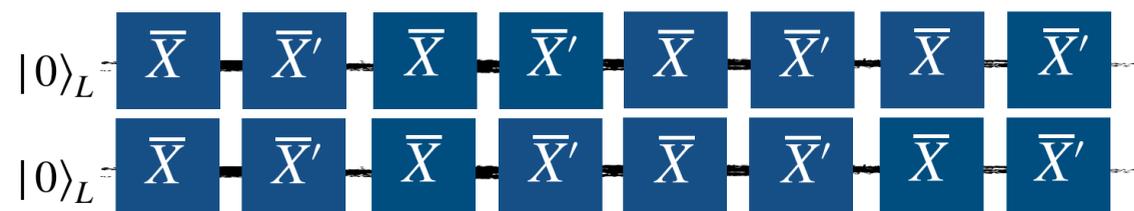
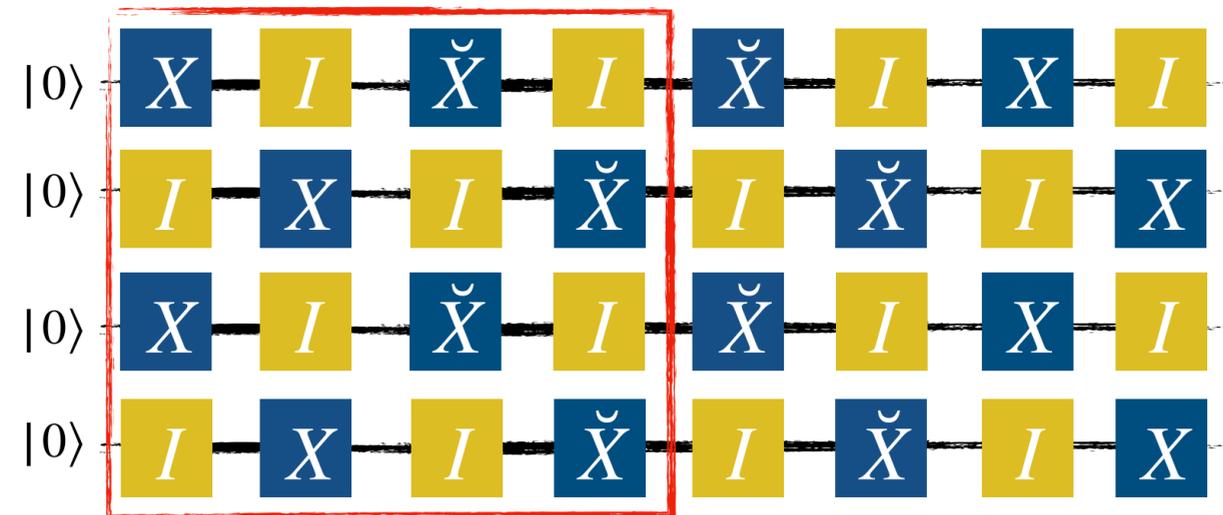


(b) Logical Error Set:  $\{\bar{I}, \bar{Z}_i, \bar{X}_i, \bar{Y}_i\}$

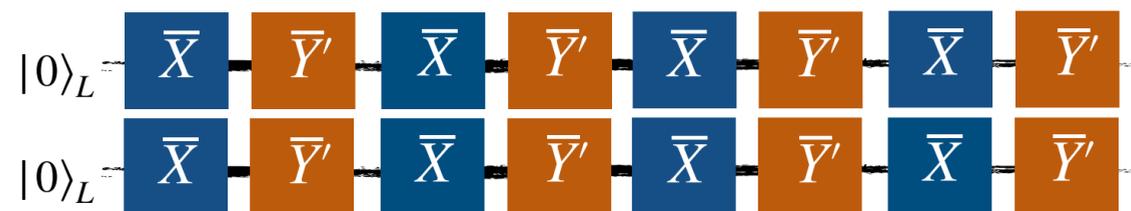
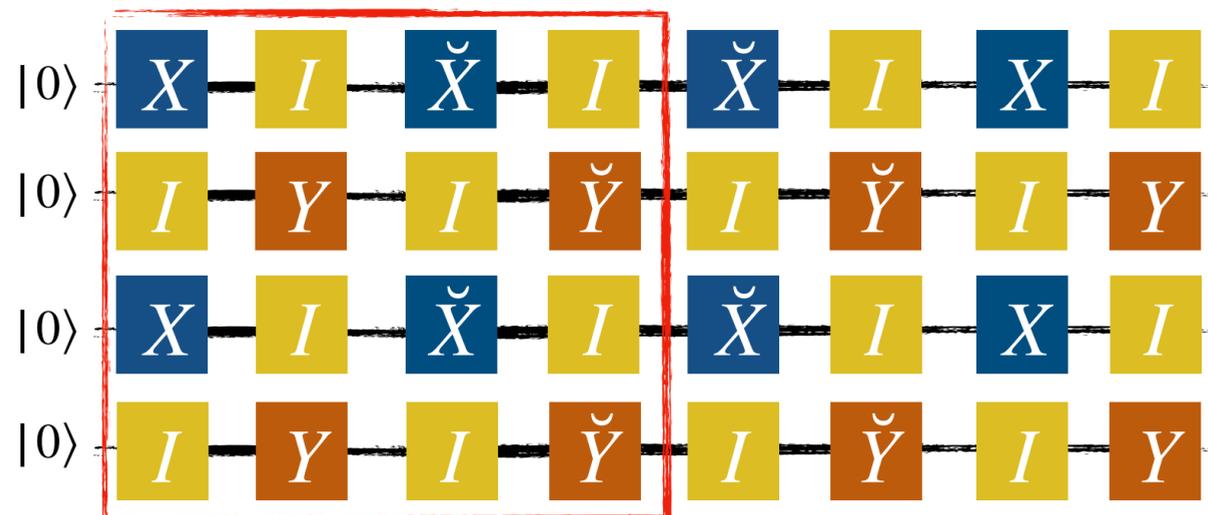
(c) 

$U_{enc,\chi}^\dagger \backslash U_{enc,\chi}$	$\Phi_+$	$\Phi_-$	$\Psi_+$	$\Psi_-$
$\Phi_+$	$\{0000, 0110, 1001, 1111\}$	$\{0000, 0110, 1001, 1111\}$	$\{0000, 0110, 1001, 1111\}$	$\{0000, 0110, 1001, 1111\}$
$\Phi_-$	$\{0000, 0110, 1001, 1111\}$	$\{0000, 0110, 1001, 1111\}$	$\{0000, 0110, 1001, 1111\}$	$\{0000, 0110, 1001, 1111\}$
$\Psi_+$	$\{0000, 0110, 1001, 1111\}$	$\{0000, 0110, 1001, 1111\}$	$\{0000, 0110, 1001, 1111\}$	$\{0000, 0110, 1001, 1111\}$
$\Psi_-$	$\{0000, 0110, 1001, 1111\}$	$\{0000, 0110, 1001, 1111\}$	$\{0000, 0110, 1001, 1111\}$	$\{0000, 0110, 1001, 1111\}$

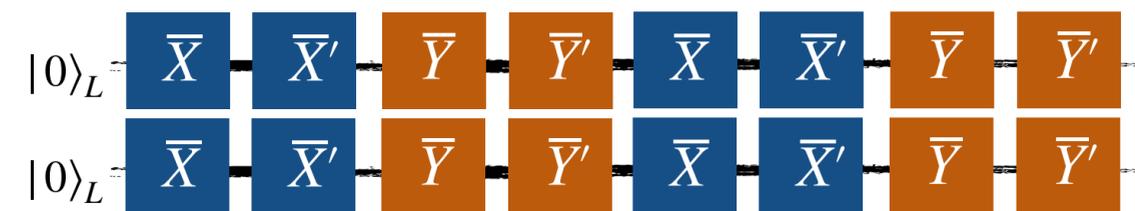
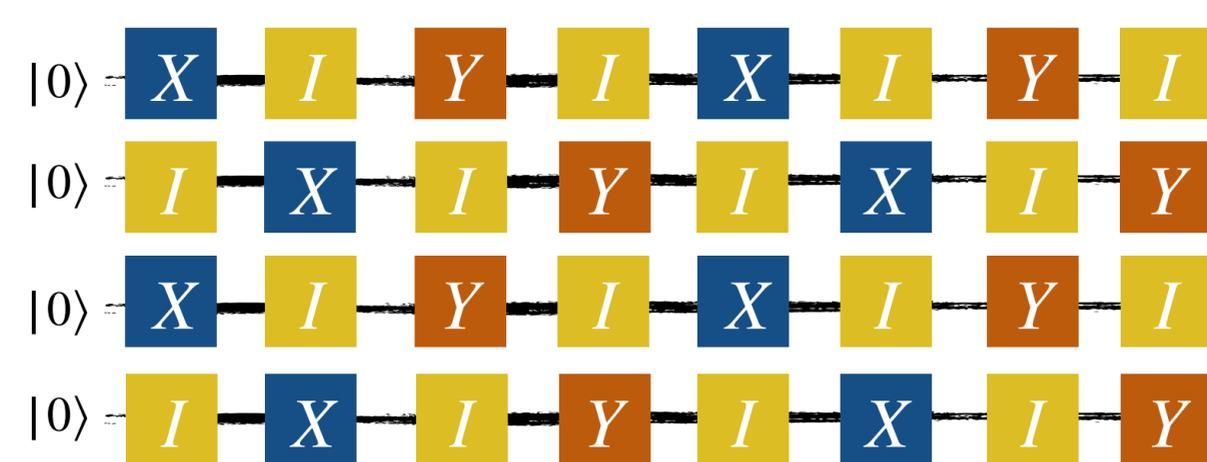
(d)

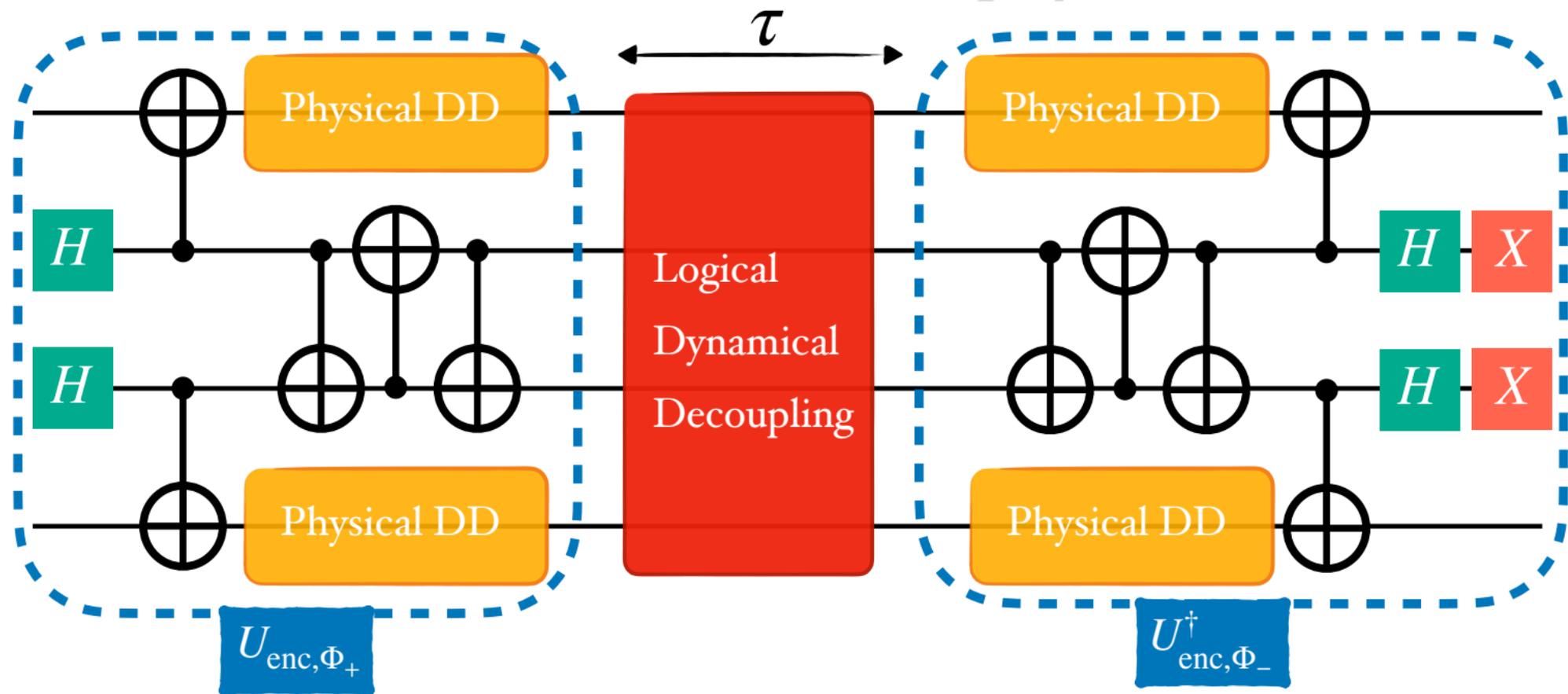


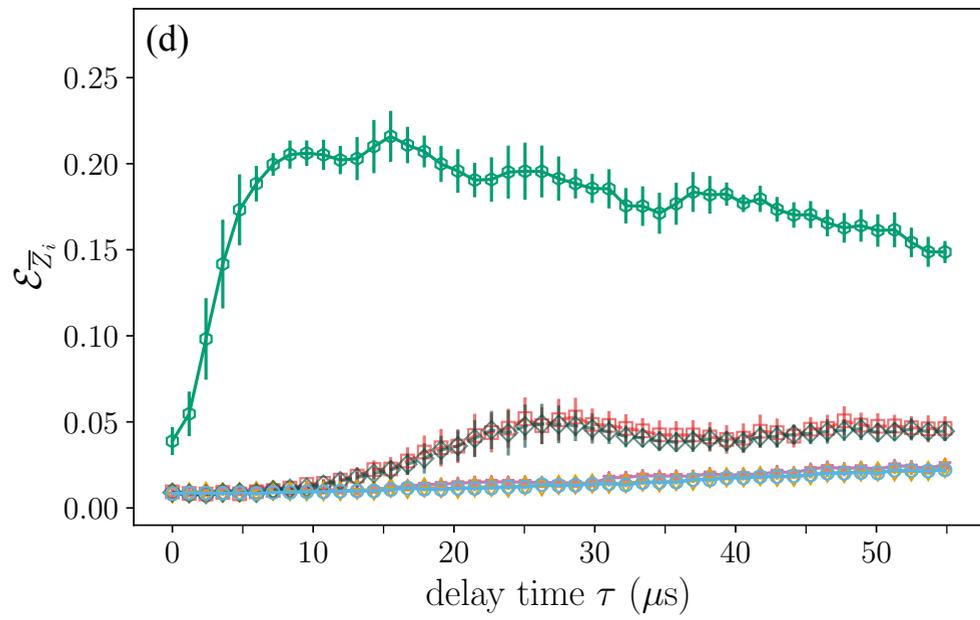
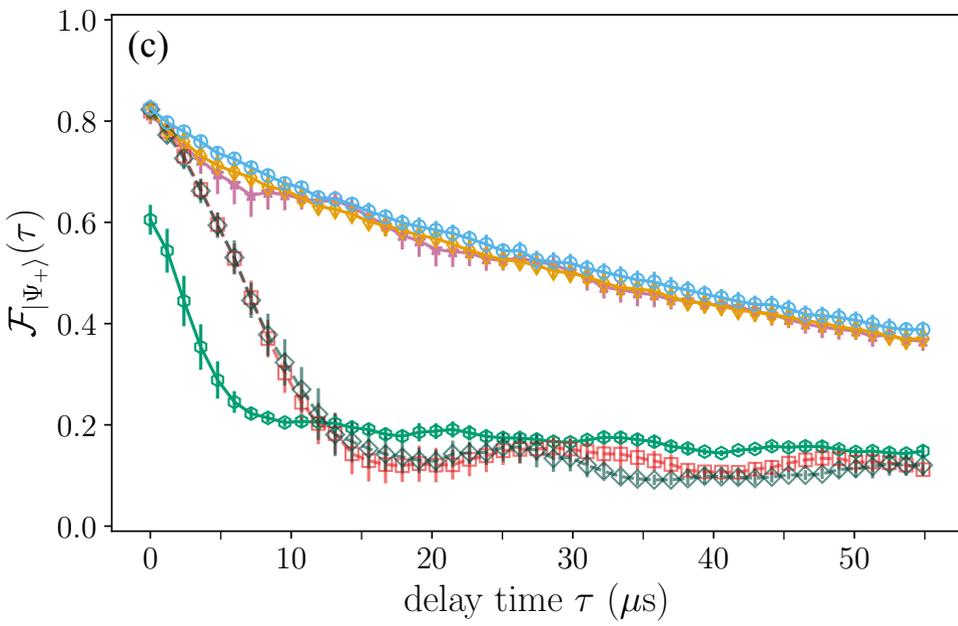
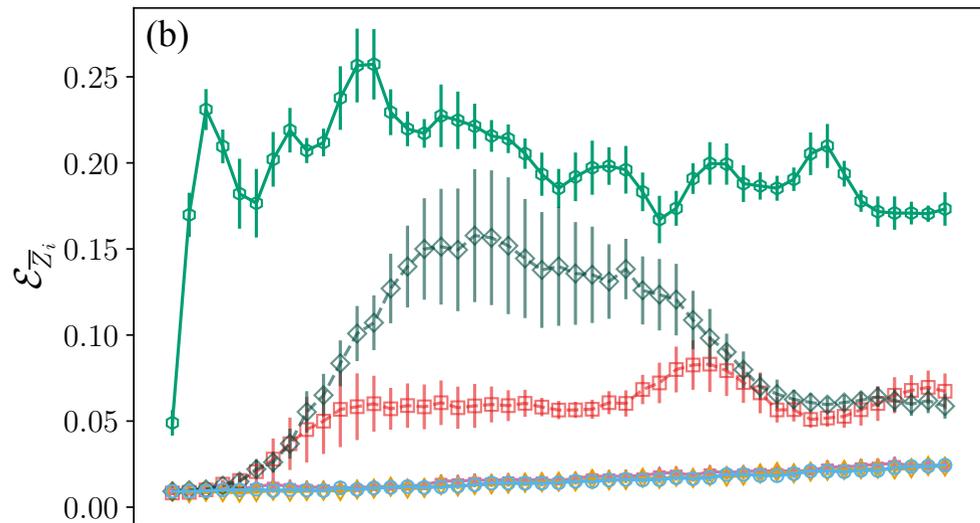
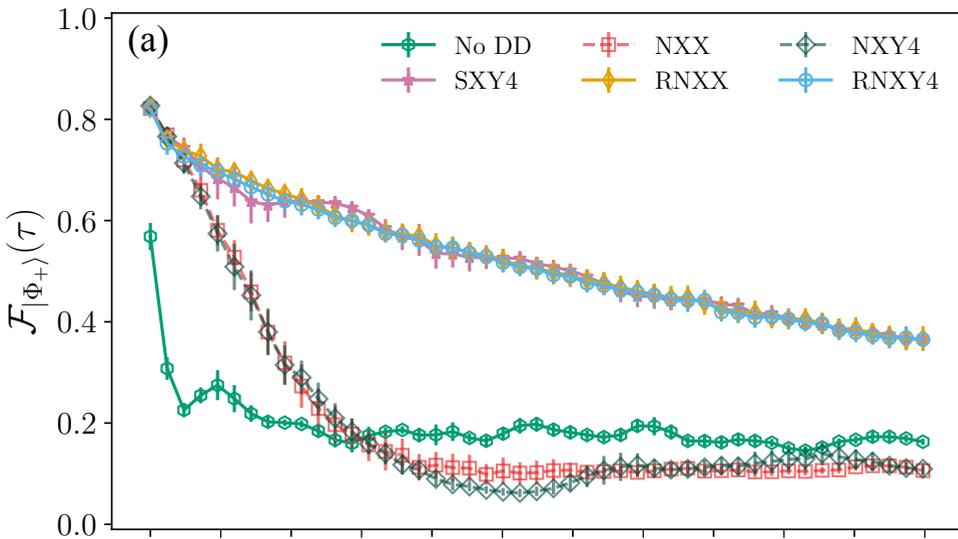
(e)

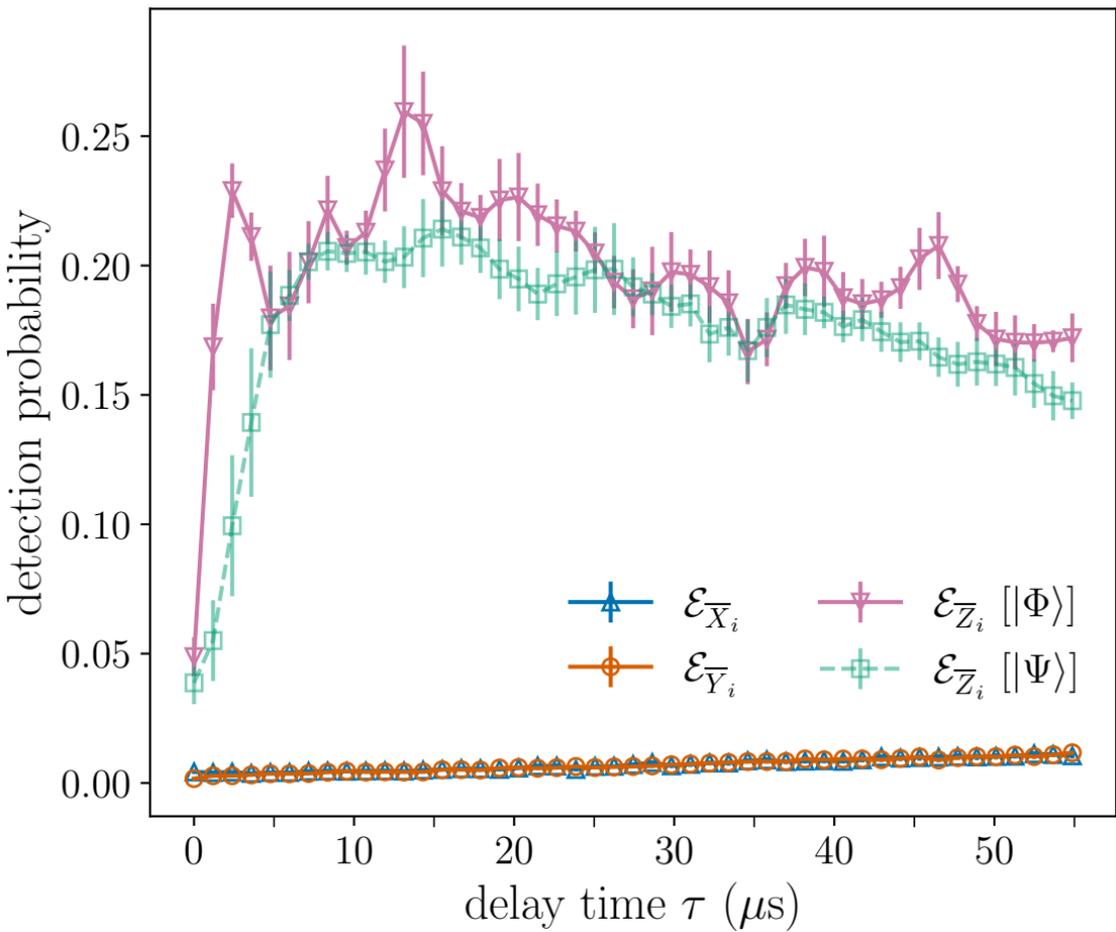


(f)









◆ Best Physical No DD    
 ◆ Best Physical SXY4    
 ◆ No DD    
 ◆ No DD+PS    
 ◆ RNXY4    
 ◆ RNXY4+PS

