Combining Error Detection and Mitigation: A Hybrid Protocol for Near-Term Quantum Simulation

Dawei Zhong,[1,2] William Munizzi,[2,3] Huo Chen,[2,4] and Wibe Albert de Jong[2]

[1] *University of Southern California*
[2] *Lawrence Berkeley National Laboratory*
[3] *University of California, Los Angeles*
[4] *Harvard University*

Practical implementation of quantum error correction is currently limited by near-term quantum hardware. In contrast, quantum error mitigation has demonstrated strong promise for improving the performance of noisy quantum circuits without the requirement of full fault tolerance. In this work, we develop a hybrid error suppression protocol that integrates Pauli twirling, probabilistic error cancellation, and the $[[n, n-2, 2]]$ quantum error detecting code. In addition, to reduce overhead from error mitigation components of our method, we modify Pauli twirling by lowering the number of Pauli operators in the twirling set, and apply probabilistic error cancellation at the end of the encoded circuit to remove undetectable errors. Finally, we demonstrate our protocol on a non-Clifford variational quantum eigensolver circuit that estimates the ground state energy of H_2 using both `qiskit` AerSimulator and the IBM quantum processor `ibm_brussels`.

I. INTRODUCTION

Quantum computing is expected to offer asymptotically faster solutions to classically difficult problems [1], particularly those in quantum chemistry [2], quantum simulation [3], cryptography [4] and optimization [5]. However, the presence and accumulation of physical errors in quantum circuits can lead to unreliable outputs, thereby preventing the realization of authentic quantum advantage. Quantum error correction codes (QECCs) [6–9] are widely considered a promising approach for reversing errors and preserving high-fidelity quantum computation. Quantum error correction works by encoding information into a subspace of the full Hilbert space, known as the codespace. When an error has occurred on the stored information, the logical state is moved outside the codespace. When such a transformation has occurred, the precise deviation is identified, and a corrective operation is applied to map the logical state back into the codespace, preserving the stored logical information. Recently, several milestone results for QECCs have been experimentally verified and presented on different hardware platforms. Some notable features of these recent demonstration include codes with below fault-tolerant threshold memory [10], logical operations performed on multiple qubits and entangled state preparation [11, 12], teleportation of logical states [13], and experimentally realized magic state distillation [14].

Alongside fault-tolerant error correction, recent advancements in error mitigation have likewise been accelerating our understanding of devices in the noisy intermediate-scale quantum (NISQ) era. Notable examples of promising error mitigation techniques include zero noise extrapolation (ZNE) [15, 16], probabilistic error cancellation (PEC) [16, 17], Pauli twirling [18], and measurement error mitigation (MEM) [19]. Each of these algorithmic protocols enhances the accuracy of measured expectation values by post-processing data from an ensemble of noisy circuits. These methods are collectively referred to as quantum error mitigation (QEM) [20], and their utility for near-term quantum computation has been rigorously demonstrated [21, 22].

Both quantum error correction and quantum error mitigation have their respective limitations. QECCs require additional qubits and gates to protect quantum information and operations, often introducing new sources of error in the process. For this reason, successful quantum error correction demands that physical error rates in quantum devices lie below a certain threshold [23] to outperform the uncorrected circuit. For contemporary quantum devices, lowering physical errors to the required regime is a demanding and non-trivial task. Moreover, it is well known that Clifford unitaries are not sufficient for universal quantum computation [9]. The fault-tolerant implementation of non-Clifford operations requires costly resources such as magic states. These operations often rely on resource-intensive protocols, including magic state distillation [24] for high-fidelity T-states, or gate teleportation techniques for implementing gates such as the CCZ [25].

Meanwhile, QEM offers a practical alternative for minimizing error on NISQ devices, due to its reduced hardware overhead compared to quantum error correction. However, since QEM inherently requires averaging over many samples, the sampling costs associated with QEM grow exponentially with the number of physical error events [26]. This feature often renders contemporary QEM protocols impractical when the circuit *size*, i.e. the circuit depth times the qubit number, becomes large [20]. Considering limitations of both QECCs and QEM, it is natural to ask whether a hybrid approach could utilize the respective advantages of each while minimizing the associated overhead. Several recent attempts have been made to investigate the possibility of a hybrid combination of QEC and QEM. For example, encoded Clifford+T circuits can be protected by QECCs, while using PEC [27, 28] to mitigate the errors in noisy T-state preparation. This approach reduces the hardware overhead of magic state distillation at the expense of an increased sampling cost. A similar technique, combining PEC with QEC, has been demonstrated within a fault-tolerant quantum computation (FTQC) architecture [29]. This integration reduces the qubit overhead required for FTQC, enabling more logical operations to be performed using the same fixed hardware resources.

In this work, we further explore the potential of hybrid QEC/QEM approaches, and develop a framework that integrates the $[[n, n-2, 2]]$ quantum error detection code (QEDC) [30, 31] with Pauli twirling and PEC. QEDC is widely considered a suitable starting point for the practical implementation of stabilizer codes [32–35] since it requires only constant qubit overhead and simple post-processing. Moreover, QEDCs offer a simpler encoding scheme for single-qubit and two-qubit logical rotations [30, 31], eliminating the need for compilation into Clifford+T circuits and avoiding circuit synthesis errors due to compilation approximation. However, there still remain a significant amount of undetectable errors in the simple, but non-fault-tolerant, encoded circuit. In an attempt to mitigation these undetectable errors, we apply Pauli twirling and PEC to eliminate remaining noise and recover noisy operator expectation values. This hybrid approach additionally reduces the sampling overhead typically required by Pauli twirling and PEC, since the use of quantum error detection lowers the level of noise to be mitigated. Accordingly, our combined approach offers mutual benefits while preserving the individual error suppression capacity for each technique.

In this paper, we first provide a background review of quantum error detection codes (QEDC) and probabilistic error cancellation (PEC), and introduce our proposed hybrid framework in Sec. II. We then review the concept of Pauli twirling and introduce our custom partial twirling protocol in Sec. III. We demonstrate our combined QEDC/PEC protocol, using both simulation as well as actual IBM quantum processors, by protecting an encoded VQE circuit for H_2 ground state energy estimation in Sec. IV. Finally, we summarize our results and discuss future applications in Sec. V.

II. COMBINATION OF QUANTUM ERROR DETECTION WITH PEC

In this section, we discuss how to integrate quantum error detection and quantum error mitigation. First, we will introduce the quantum error detection code in Sec. II A and PEC in Sec. II B. Then we describe the error detection protocol with PEC in Sec. II C. Finally, we explain how to determine the logical noise channel after post-selection in Sec. II D, which is a crucial step before utilizing PEC in our protocol.

A. Quantum Error Detection Code

The quantum stabilizer code formalism is a widely used framework for analyzing and designing quantum error-correcting codes using symmetries of quantum states. Stabilizer codes define their codespace as the joint $+1$ eigenspace of an abelian subgroup of Pauli operators, and are identified by three parameters $[[n, k, d]]$, where n is the number of physical qubits, k is the number of logical qubits, and d is the code distance. The $[[n, n-2, 2]]$ quantum error detection code (QEDC) is a stabilizer code defined by two non-local Pauli operators, $X^{\otimes n}$ and $Z^{\otimes n}$. This family of code utilizes an even number of physical qubits n to encode $k = n-2$ logical qubits with code distance 2. If labeling n-physical qubits as $\{q_x, q_z, q_k, q_{k-1}, \ldots, q_j, \ldots, q_2, q_1\}$ where j represents the j-th logical qubit, then the left circuit of Fig 1 is used to prepare an arbitrary logical state. Inverting the state preparation circuit gives the decoding circuit as shown in the right panel of Fig. 1, which changes the final encoded state of n-qubits into the corresponding unencoded state of $(n-2)$-qubits. In addition, the decoding circuit can be used to detect errors; measuring -1 in any of the first two qubits suggests that the final state is corrupted and should be discarded. The logical Pauli operators on the j-th logical qubit is defined by $\overline{X}_j = X_{q_z} X_{q_j}$, $\overline{Z}_j = Z_{q_x} Z_{q_j}$ and $\overline{Y}_j = \mathrm{i}\overline{X}_j \overline{Z}_j$. Here we use the notation $\overline{\cdot}$ for logical operations and P_{q_j} for the Pauli operator on the physical qubit with the label q_j.

To achieve universal quantum computation, a set of Clifford operations and a non-Clifford operation are required. Ref. [31] shows how to encode Hadamard, phase and CNOT gate into the

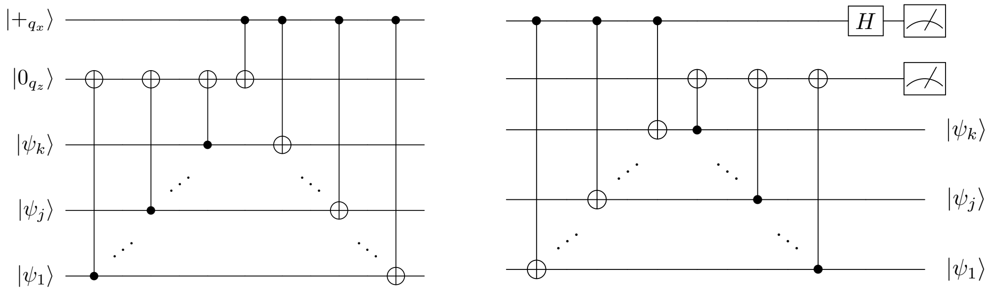

Figure 1. State preparation (left) and decoding (right) circuits for an arbitrary logical state. Note that these circuits are not fault-tolerant, but using them in this paper would not affect the main result.

$[[n, n-2, 2]]$ code, but implementing logical non-Clifford gates are usually difficult and resource consuming. Fortunately, the exponential operators, which is defined by $\exp(-\mathrm{i}\theta P)$ with P as an arbitrary n-qubit Pauli operator, can be encoded using quantum error detection codes by [30, 31]

$$\overline{\exp(-\mathrm{i}\theta P)} \xrightarrow{\text{encoded into}} \exp(-\mathrm{i}\theta \overline{P}), \tag{2.1}$$

where $\overline{e^{-\mathrm{i}\theta P}} = \cos\theta I - \mathrm{i}\sin\theta\overline{P} = e^{-\mathrm{i}\theta\overline{P}}$. Specially, the rotations on j-th logical qubits are given by [30]

$$\overline{R_{X_j}}(\theta) = \exp(-\mathrm{i}\theta\overline{X}_j/2) = \exp(-\mathrm{i}\theta X_{q_z}X_{q_j}/2), \tag{2.2}$$
$$\overline{R_{Z_j}}(\theta) = \exp(-\mathrm{i}\theta\overline{Z}_j/2) = \exp(-\mathrm{i}\theta Z_{q_x}Z_{q_j}/2). \tag{2.3}$$

In this paper, we will only use exponential maps and logical Pauli operations to implement a simple VQE algorithm (see Section IV).

The $[[n, n-2, 2]]$ code violates the quantum Hamming bound, so it cannot uniquely identify different errors and can only detect Pauli errors that anticommute with at least one of its stabilizer generators. The measurement of two stabilizer generators will tell whether an detectable error occurs and whether the data should be discarded. Note that Pauli errors that commute with all stabilizer generators are called logical errors. This type of noise will not trigger any of two syndrome measurements thus cannot be removed by the $[[n, n-2, 2]]$ code.

Some coherent errors can also be detected and removed by the code. Consider an arbitrary single-qubit error E which can be written as [36]

$$E = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \frac{a+d}{2}I + \frac{b+c}{2}X + \frac{-b+c}{2}XZ + \frac{a-d}{2}Z. \tag{2.4}$$

In this case, a noisy physical state $E|\psi\rangle$ can be written as a linear combination of $|\psi\rangle$, $X|\psi\rangle$, $XZ|\psi\rangle$, $Z|\psi\rangle$. If measuring stabilizer generators $X^{\otimes n}$ and $Z^{\otimes n}$, one will obtain $(+1,-1)$, $(-1,-1)$ or $(-1,+1)$ with probability $1 - |(a+d)/2|^2$, which results from the term $X|\psi\rangle, XZ|\psi\rangle$ or $Z|\psi\rangle$, respectively, and can be removed by post-selection. The above procedure is still valid if E is a n-qubit error, because the noisy state can still be expressed as a linear combination of a noisy state with detectable Pauli error $P|\psi\rangle$.

B. Probabilistic Error Cancellation

Probabilistic error cancelation (PEC) is designed to enhance the accuracy of the expectation value by making use of noise channel information. Assuming all gate errors are Markovian—so that errors from different layers are independent—a noisy quantum circuit with L-layer is

$$\rho = \widetilde{\mathcal{U}}(\rho_0) = \widetilde{\mathcal{U}}_L \circ \cdots \circ \widetilde{\mathcal{U}}_1(\rho_0) = \mathcal{N}_L \circ \mathcal{U}_L \circ \cdots \circ \mathcal{N}_1 \circ \mathcal{U}_1(\rho_0), \tag{2.5}$$

where $\widetilde{\cdot}$ denotes noisy quantity, ρ_0 and ρ are the input and output state, $\mathcal{U}_l(\rho) \equiv U_l \rho U_l^\dagger$ and $\mathcal{N}_l$ represents the ideal operation and the associated noise of the l-th noisy layer $\widetilde{\mathcal{U}}_l$, respectively. The expectation value of observable A measured from the above noisy experiment is

$$\langle \widetilde{A} \rangle = \mathrm{Tr}(A\,\rho) = \mathrm{Tr}\left[A\widetilde{\mathcal{U}}(\rho_0)\right]. \tag{2.6}$$

To estimate the noise-free expectation value $\langle \widehat{A} \rangle$ by PEC, one needs to identify the complete information of each $\mathcal{N}_l$, then apply the inverse of $\mathcal{N}_l$ after each layer to remove the effect of noise and finally determine $\langle \widehat{A} \rangle$. In practice, it is convenient to write $\mathcal{N}_l^{-1}$ as a linear combination of $\{\mathcal{O}_{\alpha_l}\}$, i.e., $\mathcal{N}_l^{-1} = \sum_{\alpha_l} \eta_{\alpha_l} \mathcal{O}_{\alpha_l}$. Here, $\{\mathcal{O}_{\alpha_l}\}$ is a set of implementable but noisy operation and $\{\eta_{\alpha_l}\}$ follows a quasi-probability distribution where $\sum_{\alpha_l} \eta_{\alpha_l} = 1$ but η_{α_l} can be negative. The quasiprobability distribution $\{\eta_{\alpha_l}\}$ can be transformed into a normal probability distribution $\{q_{\alpha_l}\}$ through $q_{\alpha_l} = |\eta_{\alpha_l}|/\gamma_l$ and $\gamma_l = \sum_{\alpha_l} |\eta_{\alpha_l}| \geq 1$. In this case, the l-th layer unitary is recovered from

$$\mathcal{U}_l = \mathcal{N}_l^{-1} \circ \mathcal{N}_l \circ \mathcal{U}_l = \sum_{\alpha_l} \eta_{\alpha_l} \mathcal{O}_{\alpha_l} \circ \widetilde{\mathcal{U}}_l = \gamma_l \sum_{\alpha_l} \mathrm{sgn}(\eta_{\alpha_l}) q_{\alpha_l} \mathcal{O}_{\alpha_l} \circ \widetilde{\mathcal{U}}_l \tag{2.7}$$

where $\mathrm{sgn}(\eta_{\alpha_l})$ is 1 if $\eta_{\alpha_l} > 0$ and -1 if $\eta_{\alpha_l} < 0$. The overall noiseless circuit is obtained by

$$\begin{aligned} \mathcal{U} &= \left(\mathcal{N}_L^{-1} \circ \mathcal{N}_L\right) \circ \mathcal{U}_L \circ \cdots \circ \left(\mathcal{N}_1^{-1} \circ \mathcal{N}_1\right) \circ \mathcal{U}_1 \\ &= \sum_{\alpha_L, \ldots, \alpha_1} \eta_{\alpha_L} \cdots \eta_{\alpha_1} \mathcal{O}_{\alpha_L} \circ \widetilde{\mathcal{U}}_L \circ \cdots \circ \mathcal{O}_{\alpha_1} \circ \widetilde{\mathcal{U}}_1 \\ &= \gamma \sum_{\alpha_L, \ldots, \alpha_1} \mathrm{sgn}(\eta_{\alpha_L}) \cdots \mathrm{sgn}(\eta_{\alpha_l}) q_{\alpha_L} \cdots q_{\alpha_1} \mathcal{O}_{\alpha_L} \circ \widetilde{\mathcal{U}}_L \circ \cdots \circ \mathcal{O}_{\alpha_1} \circ \widetilde{\mathcal{U}}_1. \end{aligned} \tag{2.8}$$

where $\gamma = \prod_l \gamma_l$. Instead of calculating expectation value directly from Eq. (2.8), we can creates circuits from N samples of $\{\mathcal{O}_{\alpha_1}, \ldots, \mathcal{O}_{\alpha_L}\}$ generating from probability distribution $\{q_{\alpha_1}, \ldots, q_{\alpha_L}\}$, and estimate expectation with

$$\langle \widehat{A} \rangle = \frac{\gamma}{N} \sum_k s_k \mathrm{Tr}\left[A\,\mathcal{O}_{\alpha_L} \circ \widetilde{\mathcal{U}}_L \circ \cdots \circ \mathcal{O}_{\alpha_1} \circ \widetilde{\mathcal{U}}_1(\rho_0)\right] \tag{2.9}$$

where $s_k = \prod \mathrm{sgn}(\eta_{\alpha_l})$ is the sign corresponding to the product of a given sample of $\{\mathcal{O}_{\alpha_l}\}$ [37].

Handling a general noise channel with PEC is still difficult. It is a common practice to convert a general noise channel into a stochastic Pauli noise with Pauli twirling and replace $\{\mathcal{O}_{\alpha_l}\}$ with Pauli operation $\{\mathcal{P}_{\alpha_l}\}$ as implementable noisy operations. In this paper, we will adopt this setting and utilize PEC only to cancel Pauli noise.

C. Protocol

According to Sec. II A- II B, the $[[n, n-2, 2]]$ QEDC is only able to detect all weight 1 and some weight 2 Pauli errors, while PEC can eliminate the impact of any type of noise and return a noise-free expectation. We integrate two techniques to enhance the performance of QEDC by removing logical errors with PEC. To be specific, consider a noisy encoded circuit with totally L-layers shown in Fig. 2,

$$\widetilde{\mathcal{D}} \circ \widetilde{\mathcal{U}} \circ \widetilde{\mathcal{E}}(\rho_0) = \mathcal{N}_L \circ \mathcal{U}_L \circ \cdots \circ \mathcal{N}_1 \circ \mathcal{U}_1(\rho_0), \tag{2.10}$$

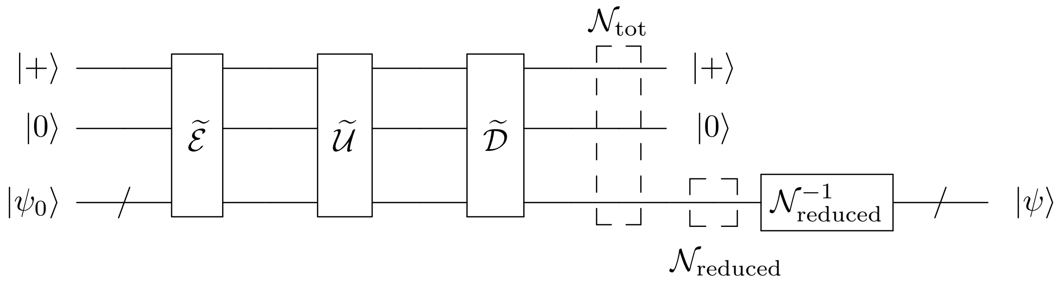
$\mathcal{N}_{\text{tot}}$
$|+\rangle$
$|0\rangle$
$|\psi_0\rangle$
$\widetilde{\mathcal{E}}$
$\widetilde{\mathcal{U}}$
$\widetilde{\mathcal{D}}$
$|+\rangle$
$|0\rangle$
$\mathcal{N}^{-1}_{\text{reduced}}$
$|\psi\rangle$
$\mathcal{N}_{\text{reduced}}$

Figure 2. The $[[n, n-2, 2]]$ error detection protocol with PEC. Here $|\psi_0\rangle$ is the input logical state. This diagram include noisy state-preparation $\widetilde{\mathcal{E}}$, noisy unitary operation $\widetilde{\mathcal{U}}$ and noisy decoding $\widetilde{\mathcal{D}}$. It also displays the "location" of equivalent noise channel $\mathcal{N}_{\text{tot}}$ and $\mathcal{N}_{\text{reduced}}$.

where $\rho_0 = |+\rangle\langle+| \otimes |0\rangle\langle 0| \otimes |\psi_0\rangle\langle\psi_0|$ and the overall protocol includes state preparation $\mathcal{E}$, unitary operation $\mathcal{U}$, and decoding circuit $\mathcal{D}$. For $[[n, n-2, 2]]$ code, we adopt the decoding circuit in Fig. 1 for $\mathcal{D}$ and use it to convert n-qubit physical state into logical state. Also, the decoding circuit will be used for error detection. We will select the data if both of the first two qubits are measured as +1. It is possible to rewrite Eq. (2.10) as

$$\widetilde{\rho} = \widetilde{\mathcal{D}} \circ \widetilde{\mathcal{U}} \circ \widetilde{\mathcal{E}}(\rho_0) = \mathcal{N}_{\text{tot}} \circ \mathcal{D} \circ \mathcal{U} \circ \mathcal{E}(\rho_0). \tag{2.11}$$

where noise channel $\mathcal{N}_{\text{tot}}$ is equivalent to all noise effects in the noisy encoded circuit. After error detection and post-selection, we have

$$\widetilde{\rho} = \widetilde{\mathcal{D}} \circ \widetilde{\mathcal{U}} \circ \widetilde{\mathcal{E}} \xrightarrow{\text{error detection and post-selection}} \mathcal{N}_{\text{reduced}}(\rho). \tag{2.12}$$

Here, $\mathcal{N}_{\text{reduced}}(\rho)$ is the noisy quantum state after post-selection, where ρ is the noiseless final state and $\mathcal{N}_{\text{reduced}}$ only contains Pauli errors that cannot be removed by the QEDC.

The implementation of our protocol is outlined below, followed by a detailed explanation of each step.

1. Create a physical circuit $\mathcal{D} \circ \mathcal{U} \circ \mathcal{E}$.

2. Perform interleaved cycle benchmarking [38] to learn associated noise for each of the L layers in the physical circuit.

3. Compute $\mathcal{N}_{\text{reduced}}$ and its inverse.

   3a. Use result from Step 2 to calculate $\mathcal{N}_{\text{tot}}$ by numerical simulation of error propagation.

   3b. Delete all Pauli terms in $\mathcal{N}_{\text{tot}}$ that anti-commutes with X at the first qubit and anti-commutes with Z at the second qubit. This step outputs $\mathcal{N}_{\text{reduced}}$.

   3c. Determine the inverse of reduced noise channel $\mathcal{N}^{-1}_{\text{reduced}}$.

4. Execute the encoded circuit with Pauli twirling and $\mathcal{N}^{-1}_{\text{reduced}}$ implemented using quasi-probability sampling.

5. Compute the noise-free expectation value $\langle \widehat{A} \rangle$ by PEC after post-selection.

In Step 1, we assume the physical circuit is built with single- and two-qubit Clifford gates as well as single-qubit rotation gates, which allow us to estimate the total noise channel with propagation of Pauli error. In Step 2, we assume that the noise of the same unitary operation remains unchanged across different experiments and different parts of same circuit. Therefore, the benchmarking experiment output is sufficient to describe the noise in the physical circuit and can be reused for the same unitary operation. Note that interleaved cycle benchmarking only learn Pauli noise channel, so Pauli twirling should be applied to physical circuit when using our protocol. More details of the interleaved cycle benchmarking experiment can be found in Appendix A.

Step 3a and 3b approximates $\mathcal{N}_{\text{reduced}}$ by firstly calculate $\mathcal{N}_{\text{tot}}$ then removing detectable error terms in $\mathcal{N}_{\text{tot}}$. We will provide more details of these steps in Sec. II D. In Step 4 and 5, the PEC estimator $\langle\widehat{A}\,\rangle$ is determined from

$$\langle\widehat{A}\,\rangle = \text{Tr}\left[A\mathcal{N}_{\text{reduced}}^{-1}\circ\mathcal{Q}\circ\widetilde{\mathcal{D}}\circ\widetilde{\mathcal{U}}\circ\widetilde{\mathcal{E}}(\rho_0)\right]. \tag{2.13}$$

where $\mathcal{N}_{\text{reduced}}^{-1}$ is implemented by the quasi-probability sampling following Eq. (2.9), and channel $\mathcal{Q}$ denotes the post-selection prcess.

D. Estimate Logical Noise for PEC

In this section, we are going to explain how to obtain $\mathcal{N}_{\text{tot}}$ and $\mathcal{N}_{\text{reduced}}$ for our protocol from circuit

$$\begin{aligned}\widetilde{\mathcal{D}}\circ\widetilde{\mathcal{U}}\circ\widetilde{\mathcal{E}} &= \left(\sum_{j_L} c_{j_L}\mathcal{P}_{j_L}\right)\circ\mathcal{U}_L\circ\cdots\circ\left(\sum_{j_1} c_{j_1}\mathcal{P}_{j_1}\right)\circ\mathcal{U}_1\\ &= \sum_{j_1,\ldots,j_L} c_{j_L}\cdots c_{j_1}\mathcal{P}_{j_L}\circ\mathcal{U}_L\circ\cdots\circ\mathcal{P}_{j_1}\circ\mathcal{U}_1,\end{aligned} \tag{2.14}$$

where noise of each layer in Eq. (2.10) becomes Pauli noise after twirling, i.e., $\mathcal{N}_l = \sum_{j_l} c_{j_l}\mathcal{P}_{j_l}$.

The first step is to get $\mathcal{N}_{\text{tot}}$ via error propagation,

$$\mathcal{V}\circ\mathcal{P}(\rho) = VP\rho PV^\dagger = (VPV^\dagger)V\rho V^\dagger(VPV^\dagger) = \mathcal{M}\circ\mathcal{V}(\rho), \tag{2.15}$$

where V is arbitrary unitary matrix and $\mathcal{M}(\rho) = (VPV^\dagger)\rho(VPV^\dagger)^\dagger$ is equivalent noise channel after error propagation. If V is a Clifford gate, $\mathcal{M}$ will still be a Pauli channel. If V is a single-qubit rotation, $VPV^\dagger$ will be a linear combination of Pauli operators (see Appendix B). By repeatedly using Eq. (2.15) to swap neighboring channels in Eq. (2.14), we will manage to separate noise from all noise-free unitary,

$$\widetilde{\mathcal{D}}\circ\widetilde{\mathcal{U}}\circ\widetilde{\mathcal{E}} = \mathcal{N}_{\text{tot}}\circ\mathcal{U}_L\circ\cdots\circ\mathcal{U}_1, \tag{2.16}$$

where

$$\mathcal{N}_{\text{tot}} = \sum_{j_1,\ldots,j_L} c_{j_L}\cdots c_{j_1}\mathcal{P}_{j_L}\circ\mathcal{M}_{j_{L-1}}\circ\cdots\circ\mathcal{M}_{j_1}. \tag{2.17}$$

Here $\mathcal{M}_{j_l}(\rho) = M_{j_l}\rho M_{j_l}^\dagger$ is equivalent noise channel of $\mathcal{P}_{j_l}$ in Eq. (2.14) after error propagation,

$$M_{j_l} = U_L\cdots U_{l+1}P_{j_l}U_{l+1}^\dagger\cdots U_L^\dagger. \tag{2.18}$$

The second step is to obtain the total logical noise channel after error detection and post-selection. It is convenient to express Eq. (2.17) in the form of $\mathcal{N}_{\text{tot}} = \sum_k p_k E_k\rho E_k^\dagger$, where the sum is over all possible $\{j_1,\ldots,j_L\}$. $p_k = c_{j_L}\cdots c_{j_1}$ and $E_k = P_{j_L}M_{j_{L-1}}\cdots M_{j_1}$. Note that $\{c_{j_l}\}$ gives a probability distribution of noise terms in $\mathcal{N}_l = \sum_{j_l} c_{j_l}\mathcal{P}_{j_l}$. As a result, $\{p_k\}$ represents a joint

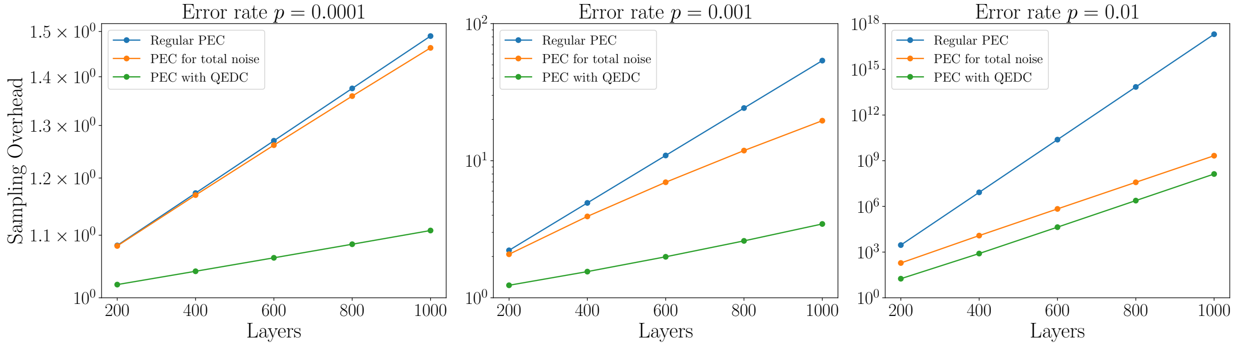

Figure 3. Sampling overhead of three error mitigation settings: PEC for layer noise, PEC for overall noise, and combined protocol with QEDC and PEC. The first two settings are evaluated on 4-qubits circuit, while the third one is evaluated on 6-qubits circuit encoded with QEDC.

probability distribution. When performing error detection on $\mathcal{N}_{\text{tot}}$, any error E_k that is a single Pauli operator and anticommutes with a stabilizer generator will be removed. If E_k is a linear combination of Pauli operators, its effects will be partially mitigated (see Appendix B for more details). Finally, the channel for all logical noise is $\mathcal{N}_{\text{logical}}(\rho) = \sum_s q_s E_s \rho E_s^\dagger$, where we renormalize $\{p_k\}$ to get $\{q_s\}$, $\rho = |\psi\rangle\langle\psi|$ is the unencoded state after noiseless $\mathcal{D} \circ \mathcal{U} \circ \mathcal{E}$ and post-selection. In this case, the noisy expectation value is derived by

$$\mathrm{Tr}\left[A\,\mathcal{N}_{\text{logical}}(\rho)\right] = \sum_s q_s \mathrm{Tr}\left[A\,E_s \rho E_s^\dagger\right]. \tag{2.19}$$

Suppose $E_s = \sum_{r_s} a_{r_s} P_{r_s}$ is a linear combination of Pauli operators, where a_{r_s} is real number and $\{a_{r_s}^2\}$ gives a probability distribution, then $E_s \rho E_s^\dagger = \sum_{r_s,k_s} a_{r_s} a_{k_s} P_{r_s} \rho P_{k_s}$ and

$$\mathrm{Tr}\left[A\,E_s \rho E_s^\dagger\right] = \sum_{\substack{r_s,k_s \\ r_s = k_s}} a_{r_s}^2 \mathrm{Tr}\left[A\,P_{r_s} \rho P_{r_s}\right] + \sum_{\substack{r_s,k_s \\ r_s \neq k_s}} a_{r_s} a_{k_s} \mathrm{Tr}\left[A\,P_{r_s} \rho P_{k_s}\right]. \tag{2.20}$$

The noisy expectation value under logical noise can be written as

$$\mathrm{Tr}\left[A\,\mathcal{N}_{\text{logical}}(\rho)\right] = \sum_{s,r_s} q_s a_{r_s}^2 \mathrm{Tr}\left[A\,P_{r_s} \rho P_{r_s}\right] + \sum_{\substack{s,r_s,k_s \\ r_s \neq k_s}} q_s a_{r_s} a_{k_s} \mathrm{Tr}\left[A\,P_{r_s} \rho P_{k_s}\right]. \tag{2.21}$$

Note that for Clifford circuit, the second term vanishes since any Pauli error will propagate through Clifford operations and become another Pauli error. For circuits with Clifford and single-qubit rotation gates, the second term is not negligible in general, it can still be addressed as a case of partial error mitigation (see Appendix B and [16, 39] for details). In this paper, we adopt the simplest strategy by neglecting the off-diagonal terms. In this case, Eq. (2.21) can be rewritten as

$$\mathrm{Tr}\left[A\,\mathcal{N}_{\text{logical}}(\rho)\right] \approx \mathrm{Tr}\left[A\,\mathcal{N}_{\text{reduced}}(\rho)\right] = \sum_j c_j \mathrm{Tr}\left[A\,P_j \rho P_j\right] \tag{2.22}$$

Here, $\mathcal{N}_{\text{reduced}} = \sum_j c_j P_j \rho P_j$ and $c_j = a_{r_s}^2 \sum_s q_s$, and its inverse can be implemented by quasi-probability sampling. To illustrate how the idea work, we will show an example in the Appendix B when E_s is linear combination of Pauli errors.

E. Overhead

The overhead of the hybrid protocol mainly comes from the sampling overhead of the PEC component, which can be evaluated by γ^2 [29] and γ is the parameter related to the quasi-probability distribution (see Sec. II B). Here, we evaluate and compare γ^2 under three error mitigation settings. The first is the regular PEC protocol that cancels noise after each layer, the second is PEC that cancels estimated overall noise, and the third is the hybrid protocol with QEDC and PEC. For the first two settings, we consider L-layers of depolarizing noise channel with the same error rate on a 4-qubit circuit. For the third setting, we consider L-layers of depolarizing noise channel but on a 6-qubit circuit, and calculate the sampling overhead of overall noise after removing detectable Pauli errors. We do not include any types of unitary operations in our simulation for simplicity. Our results for different error rates are shown in Fig. 3, which indicates that PEC on estimated overall noise has a lower sampling overhead than regular PEC protocol, and PEC on encoded circuits is more efficient than PEC on unencoded circuit. More details of the calculation can be found in the Appendix C.

III. PARTIAL TWIRLING AND PARTIAL ERROR-CORRECTION

While full fault-tolerance [40–43] remains a goal for the future, many NISQ error mitigation techniques [44–47] enable advanced noise suppression on contemporary hardware. In this section we introduce a protocol which utilizes a modified Pauli twirling scheme, in combination with partial quantum error correction, to minimize hardware noise on current quantum devices. Our technique incorporates an analysis of error propagation through Pauli twirling sets, selecting twirling operations in such a way as to randomize the errors toward a Pauli channel that closely matches the desired form. This protocol maximizes the utility of current error mitigation techniques, minimizing additional quantum resources while improving fidelity, and enhances the development of robust quantum devices and reliable information processing in the NISQ era.

A. Pauli and Partial Pauli Twirling

Pauli twirling [48–51] is an error mitigation technique which relies on conjugating a noisy channel by randomly-sampled Pauli gates to transform the noise into a depolarizing form. Pauli twirling is typically employed as a preliminary operation in many error mitigation schemes, enabling a simpler analysis, characterization, and eventual inversion of channel noise using auxiliary techniques such as probabilistic error cancellation (PEC). In this section we review Pauli twirling and describe the propagation of error through twirled quantum circuits [52]. We then introduce a modified twirling scheme, known as partial twirling, and demonstrate how it can be integrated with quantum error detection and correction. Finally, we discuss the advantages of partial twirling for improving code fidelity with reduced resource overhead, when compared to alternative approaches.

Given a unitary $\mathcal{U}$, let $\tilde{\mathcal{U}}$ denote the imperfect realization of $\mathcal{U}$ due to noise. It is useful to express $\tilde{\mathcal{U}}$ as the composition of the ideal $\mathcal{U}$ with a noise channel Λ, as

$$\tilde{\mathcal{U}} = \mathcal{U} \circ \Lambda. \tag{3.1}$$

Working in the Pauli basis, we can represent Λ as a Pauli transfer matrix which, in general, will be dense (containing many off-diagonal terms) and often non-invertible. However, through a technique known as Pauli twirling we can transform Λ into a depolarizing channel, diagonal in the Pauli basis, by conjugating Λ with randomly sampled Pauli gates $\mathcal{P}_i \in \Pi^n$ and averaging over all instances. The twirling set $\{\mathcal{P}_i\}$, i.e. the set of Pauli gates used for conjugation, is selected such that the action of $\mathcal{U}$ is unchanged. Fig. 4 illustrates the concept of Pauli twirling applied to a 2-qubit gate $\tilde{\mathcal{U}}$.

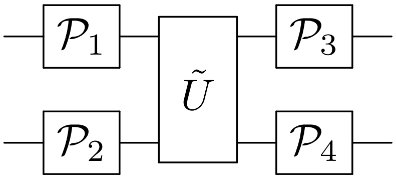

Figure 4. Pauli twirling conjugates a noisy channel $\tilde{U}$ by Pauli gates $\{P_i\}$, such that the average over conjugated channels becomes a stochastic Pauli channel. Averaging over twirled circuits converts the noise Λ, in $\tilde{\mathcal{U}}$, into a depolarizing form which can be inverted using methods such as probabilistic error cancellation.

The operation shown in Fig. 4 is repeated for all sets of Pauli gates in the twirling set $\mathcal{P} \subseteq \Pi_n$. An unbiased average is computed over all conjugated channels, building the twirled channel Λ_{twirled} as

$$\Lambda_{\text{twirled}} = \frac{1}{|\mathcal{P}|} \sum_{\mathcal{P}_i \in \mathcal{P}} \mathcal{P}_i^\dagger \Lambda \mathcal{P}_i. \tag{3.2}$$

The transformation of Λ into Λ_{twirled} converts the complex noise in $\tilde{\mathcal{U}}$, on average, into a stochastic Pauli noisy channel. As a result, the average of measured expectation values of observables is unbiased, eliminating the systematic bias introduced by accumulated coherent errors.

In this work, we combine Pauli twirling with quantum error detection (QEDC) and quantum error correction (QEC) to improve error suppression on contemporary quantum hardware [53]. We begin by introducing *partial twirling*, a modified version of the Pauli twirling protocol described above which utilizes only a subset of Pauli gates tailored to the specific QEDC/QEC scheme. Given a noisy unitary $\tilde{\mathcal{U}}$ and a QEDC/QEC protocol capable of detecting a set of errors $\mathbb{E}$, we define the partial twirling set $\tilde{\mathcal{P}}$ as the sets of Pauli gates which optimize the condition

$$\min_{\tilde{\mathcal{P}}} \left\lVert \frac{1}{|\tilde{\mathcal{P}}|} \sum_{\mathcal{P}_i \in \tilde{\mathcal{P}}} \mathcal{P}_i^\dagger \tilde{\mathcal{U}} \mathcal{P}_i - \mathbb{E} \circ \mathcal{U} \right\rVert. \tag{3.3}$$

Eq. (3.3) formulates an optimization problem over subsets of Pauli operators, seeking those whose action on $\tilde{\mathcal{U}}$ best aligns the resulting errors with the target error space $\mathbb{E}$ defined by the QEDC/QEC scheme. This ensures that, after twirling, propagated errors are optimally approximated by elements of $\mathbb{E}$. While the expression in Eq. (3.3) cannot generically be minimized to zero, there are numerous cases where it does vanish, including instances where $\mathbb{E}$ consists solely of single-qubit Pauli errors, which are invariant under the twirling action. Mathematica code to generate partial twirling sets is publicly available at [54].

When combined with partial error detection or correction, where only errors of a particular type are identifiable, partial twirling allows the choice of twirling gates to be customized to minimize undetectable errors. Moreover, the selection of twirling gates can further be adapted to the constraints of a specific hardware platform, favoring operations that are more resource-efficient or less prone to error. In this way, partial twirling provides a flexible mechanism for optimizing error suppression strategies. In the following section, we demonstrate how to integrate partial twirling with QEDC/QEC, improving the effectiveness of partial error detection and correction on NISQ hardware. We further explain how partial twirling can be implemented with reduced overhead, enabling a more efficient use of quantum resources.

B. Partial Twirling with Partial Error Correction

In the previous section, we reviewed Pauli twirling and introduced our partial twirling protocol, which strategically selects twirling gates tailored to a particular error detection/correction code or specific hardware implementation. We now demonstrate how partial twirling can be combined with error correction/detection to achieve improved error mitigation on noisy devices. We benchmark the effectiveness of partial twirling for reliably implementing IBM's native multi-qubit gates on real devices. In later sections, we extend this approach by combining partial twirling with the $[[4,2,2]]$ quantum error detection code to estimate the ground state energy of H_2 on noisy hardware.

A general schematic for combining twirling with error correction is given by the circuit diagram in Fig. 5. We begin by determining an error correction code QEC to implement, and subsequently encode the k-qubit logical state into n physical qubits using an encoding unitary $\mathcal{E}$. The goal is to evolve the logical state through some noisy unitary $\tilde{\mathcal{U}}$, representing a quantum circuit to be executed on a quantum machine, with maximum fidelity. Partial twirling is performed layer-by-layer on $\tilde{\mathcal{U}}$, indicated by the twirling blocks $\mathcal{P}$ in Fig. 5 (only one block is shown for clarity), with twirling gates $\mathcal{P}$ selected according to Eq. (3.3). After twirling and error-correction, indicated by QEC in the diagram, are performed the state is passed through a decoding operation $\mathcal{D}$. We importantly note that partial twirling does not, in general, guarantee a full conversion of the noise channel into a stochastic Pauli channel, since only a reduced subset of Pauli gates is utilized. The resulting channel depends on the specific structure of the underlying noise. Partial twirling does, however, suppress (and in in some cases completely eliminate) off-diagonal elements in the Pauli transfer matrix.

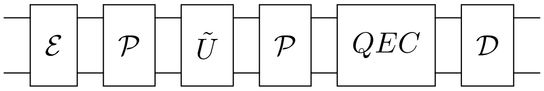

Figure 5. Combination of Pauli twirling with quantum error correction (QEC). A logical state is encoded using $\mathcal{E}$, and evolved through the noisy unitary $\tilde{\mathcal{U}}$. Twirling is implemented with operations $\mathcal{P}$ on either side of $\tilde{\mathcal{U}}$, following which the QEC is applied. The system is decoded with $\mathcal{D}$.

When QEC in Fig. 5 is a partial error correction code, only a portion of the logical code space is preserved, thereby reducing the quantum resources necessary to implement. As an example, consider the $[[3,1,1]]$ bit-flip code which can detect and reverse single-qubit Pauli X error. Fig. 6 illustrates how to combine twirling with the bit-flip code, first encoding a 3-qubit logical GHZ state into 5 physical qubits using Hadamard and CNOT gates, and appending two ancillae to perform syndrome measurement. The state GHZ is to be evolved through the noisy channel $\tilde{\mathcal{U}}$. We twirl $\tilde{\mathcal{U}}$ using the set of gates $\tilde{\mathcal{P}}$, determined by Eq. (3.3), which commute Pauli X errors through the circuit where they are detected by the bit-flip code. Fig. 6 provides a simple example of combining

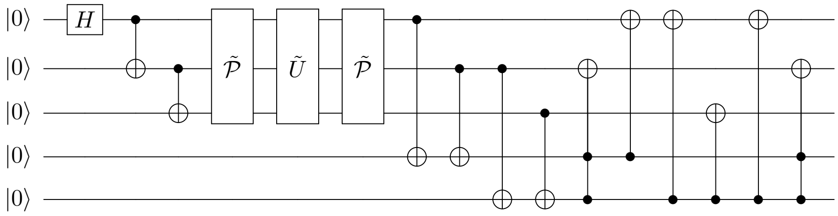

Figure 6. Circuit illustrating how to combine partial twirling with the $[[3,1,1]]$ bit-flip code with two ancillary qubits. Twirling gates $\tilde{\mathcal{P}}$ are chosen using Eq. (3.3) to commute X errors through the twirling protocol, where they can be corrected by the bit-flip code.

partial twirling with error correction. In more advanced implementations, partial twirling can be further employed to restrict twirling operations to gates most easily applied on a chosen hardware platform, or to avoid the dominant noise processes associated with a particular error model.

We use cycle benchmarking to assess the performance of our combined partial twirling with $[[3,1,1]]$ code, implemented by the circuit in Fig. 7. As before, we begin with a 5-qubit system and initialize the first three qubits in a GHZ state. We then apply an operation S, chosen from the stabilizer group of GHZ in Π^3, i.e. $S \in \{XXX,\ IZZ\}$. The sequence $\tilde{\mathcal{P}} \circ \tilde{\mathcal{U}} \circ \tilde{\mathcal{P}}$ is applied kn times, where $\mathcal{U}^k = \mathbb{I}$ and $n \in \mathbb{N}$, such that a perfect application of $\mathcal{U}$ does not modify the encoded state. After twirling $\tilde{\mathcal{U}}$ at each layer, and evolving the encoded state through the sequence of twirled $\tilde{\mathcal{U}}$ operations, the bit-flip code is applied and the expectation values of S are measured and compared for increasing depth n.

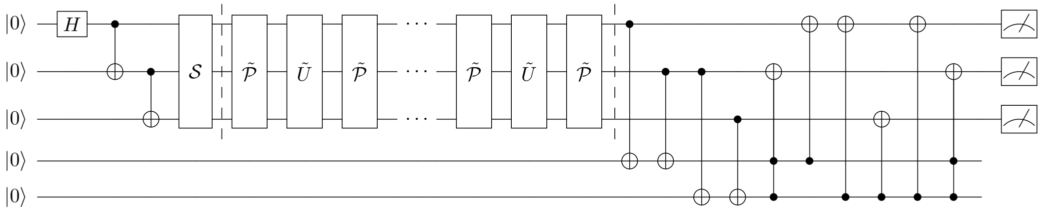

Figure 7. Cycle benchmarking a noisy unitary $\tilde{\mathcal{U}}$, using partial twirling and bit-flip code. A 5-qubit register is initialized in a GHZ state on 3 qubits, and stabilizer $\mathcal{S}$ is applied. The system is evolved through kn twirled copies of $\tilde{\mathcal{U}}$, such that $\mathcal{U}^k = \mathbb{I}$ and $n \in \mathbb{N}$. The fidelity of $\mathcal{S}$ is measured for increasing n.

We utilize the technique outlined in Fig. 7 to conduct cycle benchmarking experiments on IBM's multi-qubit gate set, using IBM quantum hardware. As an example, with results shown in Fig. S6, we benchmark the unitary

$$\mathcal{U} = CX_{1,2}CX_{2,3}, \tag{3.4}$$

the composition of 2-qubit CX gates acting on our encoded 3-qubit GHZ state. We apply partial twirling on $\tilde{\mathcal{U}}$, selecting twirling gates that cause Eq. (3.3) to vanish. We compute the fidelity of stabilizers XXX and IZZ, as a function of increasing layers of twirled $\tilde{\mathcal{U}}$. Performances are compared between no twirling, full Pauli twirling using all I, X, Y, Z combinations, and partial twirling using only I, X, Z gates. We observe in Fig. S6 that partial twirling retains higher fidelity over increasing layers of noisy $\tilde{\mathcal{U}}$ application, compared to no twirling and full Pauli twirling. Restricting to operations that commute X errors through the twirling set improves the effectiveness

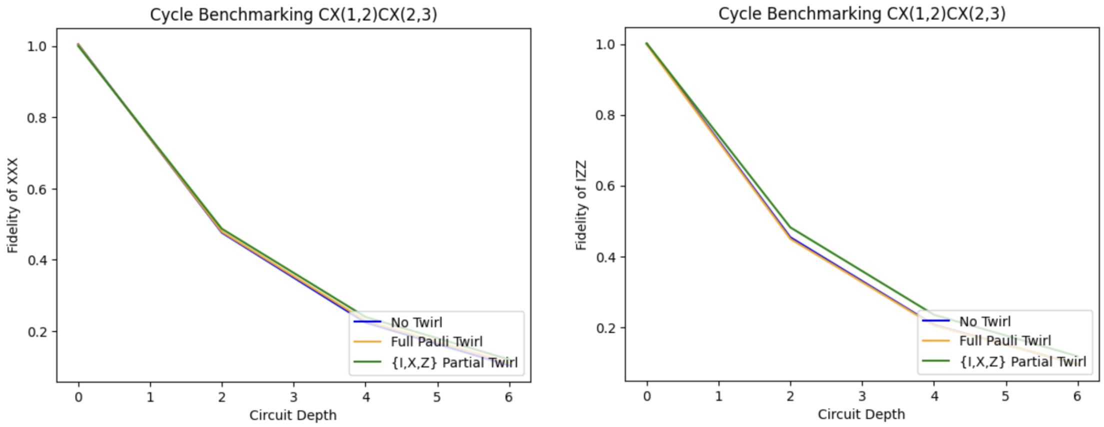

Figure 8. Cycle benchmarking of $\tilde{\mathcal{U}} = CX_{1,2}CX_{2,3}$ using twirling with bit-flip error correction. Experiment conducted on "ibm brussels" device. Fidelity of stabilizers XXX and IZZ are plotted against increasing layers of $\tilde{\mathcal{U}}$. For partial twirling, gates are selected such that Eq. (3.3) vanishes. Performance is compared between no twirling, full Pauli twirling with I, X, Y, Z gates, and partial twirling with I, X, Z gates.

of the bit-flip code in within this protocol. Moreover, by restricting to a subset of Pauli gates in the twirling step, we reduce the resource overhead needed to perform the twirling protocol (see section III D). Cycle benchmarking results for additional IBM gates, i.e. CZ, ECR, and iSWAP, are provided in Appendix D.

In this section, we demonstrated how partial twirling can be combined with partial error correction to enhance error suppression on noisy quantum hardware. We conducted an experiment, on real quantum hardware, using the $[[3, 1, 1]]$ bit-flip code with partial Pauli twirling. Twirling gates were selected to enforce a vanishing Eq. (3.3), and performance was evaluated using cycle benchmarking. In the following section, we analytically derive the logical error rate of the $[[4, 2, 2]]$ error detection code, under a noise channel, when twirling is applied in combination with error detection. This $[[4, 2, 2]]$ QEDC is then employed in Section IV to estimate the ground-state energy of H_2. Finally, we highlight the resource advantages of partial twirling for near-term error mitigation on specific hardware architectures.

C. Logical Fidelity of Twirling with Error Detection

When combining randomized compiling with quantum error detection/correction, an important consideration is the effect of channel noise on the logical error rate. In this section we analytically derive the logical fidelity of the $[[4, 2, 2]]$ error detection code [55, 56] under a noise process. We consider both the application of Pauli twirling and partial twirling, and likewise derive their respective effects on preserve the logical fidelity of our code.

We first consider a unitary noise process, characterized by an over-rotation about the Z direction, instantiate by the channel

$$\mathcal{E}\left(\rho\right) \equiv R_Z\left(\omega\right)\rho R_Z\left(-\omega\right), \qquad \text{with} \quad R_Z\left(\omega\right) = \cos(\omega/2)I + i\sin(\omega/2)Z \tag{3.5}$$

On a system of n qubits, $\mathcal{E}\left(\rho\right)$ acts independently on each qubit, and therefore the effect of this noise channel on an encoded logical state $\overline{\rho}$ is

$$\mathcal{E}^{\otimes n}\left(\overline{\rho}\right) \equiv R_Z^{\otimes n}\left(\omega\right)\overline{\rho}R_Z^{\otimes n}\left(-\omega\right). \tag{3.6}$$

The $[[4,2,2]]$ error detection code is a 4-qubit CSS code, with 2-qubit logical codeword basis given by

$$\begin{aligned} |\overline{00}\rangle &= \frac{1}{\sqrt{2}}\left(|0000\rangle + |1111\rangle\right), \\ |\overline{01}\rangle &= \frac{1}{\sqrt{2}}\left(|0011\rangle + |1100\rangle\right), \\ |\overline{10}\rangle &= \frac{1}{\sqrt{2}}\left(|0101\rangle + |1010\rangle\right), \\ |\overline{11}\rangle &= \frac{1}{\sqrt{2}}\left(|0110\rangle + |1001\rangle\right), \end{aligned} \tag{3.7}$$

The stabilizer group $\mathcal{S}$ that leaves $\{|\overline{00}\rangle, |\overline{01}\rangle, |\overline{10}\rangle, |\overline{11}\rangle\}$ invariant is generated the two Pauli strings

$$\mathcal{S} = \langle XXXX,\ ZZZZ\rangle. \tag{3.8}$$

We determine the effect of randomized compiling on code performance by computing the contribution to the code's logical fidelity [53, 57] from the noise process $\mathcal{E}^{\otimes n}(\overline{\rho})$. For a fixed choice of operator basis $\{E_m\}$, e.g. the Pauli basis, we can express a quantum channel $\mathcal{E}(\rho)$ as the sum

$$\mathcal{E}(\rho) = \sum_{m,n} \chi_{m,n} E_m \rho E_n^\dagger, \tag{3.9}$$

where χ denotes the Chi-matrix representation [58], a reshaped presentation of the Choi matrix, for the channel. Using Eq. (3.9), we construct the Chi-matrix operator for the logical noise channel $\mathcal{E}^{\otimes n}(\overline{\rho})$, which we denote $\overline{\chi}$. Given an error correction code capable of detecting a set of errors $\mathcal{E}_C$, the diagonal elements of $\overline{\chi}$ contain the correction probabilities associated with each $E \in \mathcal{E}_C$. The total probability of correctable errors p_c is then computed [53, 57] as the indexed sum over diagonal elements in $\overline{\chi}$, specifically

$$p_c = \sum_{E\in\mathcal{E}_C} \chi_{E,E}. \tag{3.10}$$

Consequently, the total probability of uncorrectable errors p_u is given by $p_u = 1 - p_c$. The value of p_u relates to the randomized benchmarking infidelity r as

$$p_u = r - \sum_{\substack{E\in\mathcal{E}_C \\ E\neq\mathbb{I}}} \chi_{E,E}. \tag{3.11}$$

For a noise channel decomposable into Pauli noise processes, p_u computes the logical infidelity $\overline{r}$ of the code under the noisy process [53, 57]. Thereby we calculate the average logical infidelity $\overline{r}$ of our $[[4,2,2]]$ code, under $\mathcal{E}^{\otimes n}(\overline{\rho})$, using

$$\overline{r} = p_u - \sum_{\substack{E,E'\in\mathcal{E}_C,\ E\neq E' \\ s(E)=s(E'),\ \overline{E}=\overline{E'}}} \chi_{E,E'}, \tag{3.12}$$

where E and E' are different errors in $\mathcal{E}_C$ that produce equivalent syndromes, i.e. $s(E) = s(E')$, and admit the same overall effect on the logical data, i.e. $\overline{E} = \overline{E'}$.

Since Pauli and partial twirling transform the matrix χ, the mitigative effects of twirling on the logical infidelity of our code can be derived from Eq. (3.12). As a function of rotation parameter ω, as in Eq. (3.6), the logical fidelity of the [[4, 2, 2]] code under $\mathcal{E}^{\otimes n}(\overline{\rho})$ decays as

$$\overline{r}_{\text{Untwirled}} = 1 - 6\cos\left(\frac{\omega}{2}\right)^6 \sin\left(\frac{\omega}{2}\right)^2 . \tag{3.13}$$

Similarly computing $\overline{r}$ for the case where full Pauli twirling is applied, we have a logical infidelity

$$\overline{r}_{\text{Full Twirling}} = 1 - 4\cos\left(\frac{\omega}{2}\right)^6 \sin\left(\frac{\omega}{2}\right)^2 . \tag{3.14}$$

Finally, for the case where we apply partial twirling using sets composed of $\{I, X, Z\}$ we find that the logical infidelity of [[4, 2, 2]] becomes

$$\overline{r}_{\text{Par. Twirling}} = 1 - 4\cos\left(\frac{\omega}{2}\right)^6 \sin\left(\frac{\omega}{2}\right)^2 - \frac{1}{288}\csc\left(\frac{\omega}{2}\right)^4 \sin(\omega)^6 . \tag{3.15}$$

Figure 9 illustrates the three expressions for logical infidelity given by Eqs. (3.13)–(3.15), plotted against increasing rotation error ω. As shown in Figure 9, both partial and full twirling significantly improve the [[4, 2, 2]] code fidelity in the presence of the noise channel $\mathcal{E}^{\otimes n}(\overline{\rho})$. Full Pauli twirling, where twirling sets composed of all Pauli gates are sampled and averaged over, performs best since the χ matrix is rendered diagonal after the twirling protocol is applied. While partial twirling does not fully diagonalize the χ matrix of the error channel, it nevertheless significantly reduces the magnitude of off-diagonal elements. This reduction offers mitigating effects, with a significant reduction in sampling during the twirling stage.

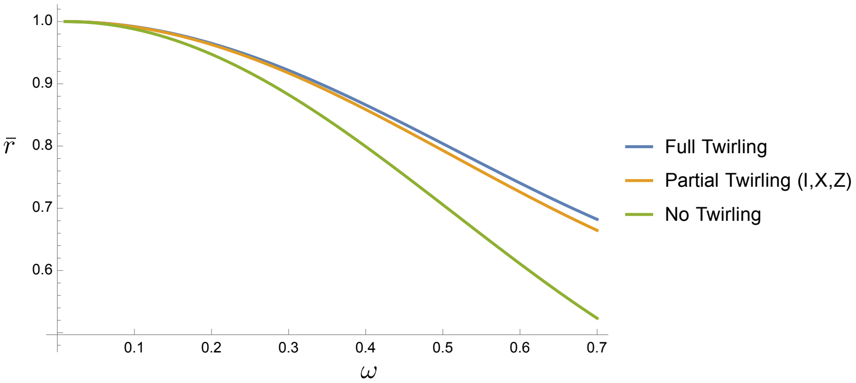

Figure 9. Logical infidelity of the [[4, 2, 2]] code under the noisy process given by Eq. (3.6), plotted against increasing error ω. We consider the case when no twirling is applied on the noisy channel, when full Pauli twirling is applied, and when partial twirling using the reduced gate set $\{I, X, Z\}$ is applied.

In this section we explained how to compute the logical infidelity of a quantum error correcting code, under a noisy process. We evaluated the logical infidelity of the [[4, 2, 2]] error detection code subject to Z rotation noise, comparing performance with and without Pauli twirling applied to the noise channel. We demonstrated that partial twirling is sufficient to retain code fidelity in the presence of noise, though it performs slightly worse than full Pauli twirling. In the following section, we highlight resource advantages of partial twirling and identify additional benefits of our hybrid error mitigation strategy.

D. Advantages and Overhead of Partial Twirling

A concern when utilizing any error mitigation scheme is the associated resource overhead, which impacts both the efficiency and feasibility of practical implementation. While contemporary error mitigation methods are designed to improve circuit fidelity, this improvement often comes with exponential sampling cost [26]. Accordingly, minimizing resource overhead is essential for any viable error mitigation strategy. For the case of Pauli twirling, one source of overhead arises from the insertion of single-qubit Pauli gates before and after the noisy unitary to be twirled. The cost of applying these single-qubit operations varies by platform, and depends heavily on hardware design and physical qubit architecture. That being said, single-qubit gates are extremely fast and high-fidelity (with error rates often around 10^{-6}) in many current hardware architectures, making the cost of implementing Pauli gates effectively negligible.

One significant advantage of partial twirling is realized on hardware platforms with a strong error bias, where certain error types occur more frequently than others. A specific example is the cat qubit [59–62], implemented using superconducting microwave cavities, where devices are engineered to intentionally suppress errors along a particular direction. In this setting, partial twirling allows users to restrict to gates which can be realized with minimal error. When combined with error correction, this approach further enables efficient error suppression with reduced resource overhead.

Another advantage of partial twirling arises in platforms where certain gates are easier to implement than others. For example, in many superconducting transmon qubit systems, a Pauli X gate (which can be executed via a simple microwave pulse) is typically very fast and reliable, whereas a Pauli Y gate might require a different calibration and could be slightly less accurate; Pauli Z rotations are often implemented as frame shifts (software updates) with effectively no physical error. In such a scenario, partial twirling makes it possible to favor the high-fidelity gates in the twirling set. One could twirl using only I, X, Z operations, for instance, avoiding Y gates if they are known to be error-prone. A smaller twirling set composed of higher-fidelity operations reduces the chance that the twirling procedure itself introduces additional errors [63].

When incorporating Pauli twirling within an error mitigation protocol, such as probabilistic error cancellation (PEC), one primary concern is the cost of sampling overhead. Sampling overhead describes the cost associated with executing numerous circuits, or *shots*, to accurately characterize the desired channel. Recall that Pauli twirling relies on sampling numerous twirled circuits to convert a noisy operation into the form

$$\Lambda_{\text{twirled}} = \sum_{i}^{N} c_i \mathcal{P}_i^{\dagger} \Lambda \mathcal{P}_i, \tag{3.16}$$

where N counts the number of twirled circuits being sampled over and c_i the appropriate coefficient for each twirled instance. When all Pauli gates are used for twirling, and all twirled circuits are sampled, N scales as 4^n for an n-qubit experiment.

Conversely, since partial twirling only ever selects a subset of Π^n to twirl with, the number of shots N is reduced and the sampling cost to implement error mitigation is lessened. Recent publications have shown that reduced twirling sets, such as those implemented in our partial twirling protocol, are capable of providing a comparable improvement to average logical fidelity when compared to twirling with the complete group of operators [63]. We importantly note, however, that many recent error mitigation schemes will choose to only sample a fixed N number of circuits, regardless of the gate set used in the Pauli twirling step. The value of N is typically chosen such that an average over N twirled circuits approximates a stochastic Pauli channel, though the channel is not exact. For the experiments performed in Section IV of this paper, we too impose a cutoff

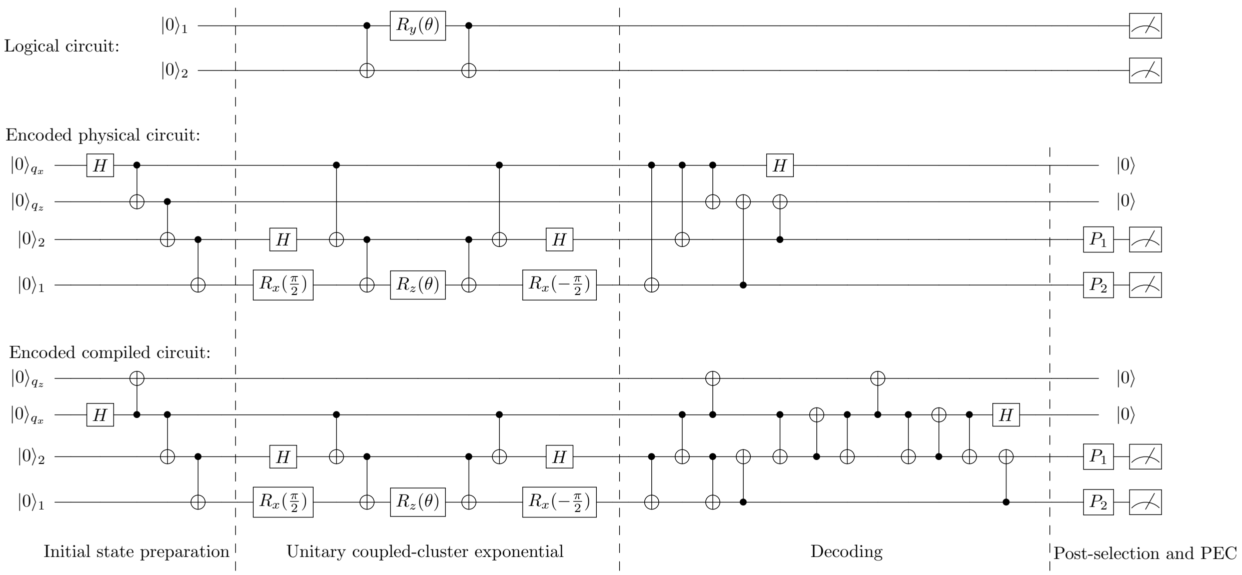

Figure 10. Quantum circuits that implement the VQE algorithm for H_2 ground state energy estimation. Encoding the logical circuit (top) using the [[4, 2, 2]] code leads to the physical circuit (middle) with initial state preparation $\mathcal{E}$, unitary operation $\mathcal{U}$ and decoding $\mathcal{D}$. The compiled circuit (bottom) is transformed from the physical circuit by swapping the first two-qubit and replacing long-range CNOT gates with equivalent neighboring CNOT gates.

on the number of circuits sampled, instead of sampling over the full group generated by our chosen twirling operators.

In this section, we introduced an error suppression protocol that integrates Pauli twirling with quantum error correction and detection. We introduced the technique of partial twirling, where a subset of twirling operations is selected to best align with the capabilities of a chosen error correction scheme, as well as the particular constraints of a given hardware platform. As a concrete example, we implemented partial twirling in combination with the $[[3, 1, 1]]$ bit-flip code on five qubits, selecting twirling operations which commute with propagating Pauli X errors. Using this hybrid method, we performed cycle benchmarking on IBM's native two-qubit gates to evaluate performance in the presence of real device noise. We then analytically derived the logical infidelity of the $[[4, 2, 2]]$ error detection code, under Z rotation noise, comparing the mitigative effects of full Pauli twirling and partial twirling. Our analysis highlighted the resource advantages gained by restricting the twirling set to a subset of Pauli gates, as well as by restricting error correction to a subspace of the logical code space. In the next section, we apply our hybrid protocol with the $[[4, 2, 2]]$ code to estimate the ground-state energy of H_2 on noisy quantum hardware, demonstrating improved accuracy and reduced sampling cost when compared to other contemporary methods.

IV. EXPERIMENT

In this section, we will demonstrate the hybrid protocol of QEDC and PEC with partial Pauli twirling on a 4-qubit VQE circuit that estimates the ground-state energy of H_2. We first provide a detailed setting of the algorithm in Sec. IV A, then verify the protocol on the simulator in Sec. IV B, and finally show both expectation values and the ground state energy evaluated from the IBM quantum processor `ibm_brussels` in Sec. IV C.

A. Experimental Setting

Our experiment adopts the smallest QEDC, which maps an arbitrary two-qubit state into a subspace of four-qubit Hilbert space as follows.

$$\begin{aligned} |\overline{00}\rangle =& \frac{1}{\sqrt{2}}(|0000\rangle + |1111\rangle), |\overline{01}\rangle = \frac{1}{\sqrt{2}}(|0110\rangle + |1001\rangle), \\ |\overline{10}\rangle =& \frac{1}{\sqrt{2}}(|0101\rangle + |1010\rangle), |\overline{11}\rangle = \frac{1}{\sqrt{2}}(|0011\rangle + |1100\rangle). \end{aligned} \tag{4.1}$$

The stabilizer generators of the $[[4, 2, 2]]$ code are $X_1X_2X_3X_4$ and $Z_1Z_2Z_3Z_4$. The logical Pauli operators for encoded state in eq. (4.1) are given by

$$\overline{X}_1 = X_2X_4, \overline{X}_2 = X_2X_3, \overline{Z}_1 = Z_1Z_4, \overline{Z}_2 = Z_1Z_3, \overline{Y}_1 = i\overline{X}_1\overline{Z}_1, \overline{Y}_2 = i\overline{X}_2\overline{Z}_2. \tag{4.2}$$

We follow the settings in Ref. [32] for our experiments. The transformed Hamiltonian of H_2 molecule is given by

$$H = g_1 + g_2Z_1 + g_3Z_2 + g_4Z_1Z_2 + g_5X_1X_2. \tag{4.3}$$

The VQE algorithm solves for the ground state by searching for the parameter θ in the unitary coupled-cluster (UCC) ansatze, $|\psi(\theta)\rangle = e^{-i\theta Y_1X_2/2}|00\rangle$, which minimizes the energy

$$E(\theta) = \langle\psi(\theta)|H|\psi(\theta)\rangle = g_1 + g_2\langle Z_1\rangle_\theta + g_3\langle Z_2\rangle_\theta + g_4\langle Z_1Z_2\rangle_\theta + g_5\langle X_1X_2\rangle_\theta. \tag{4.4}$$

The quantum circuit that implements the UCC ansatze is shown at the top of Fig. 10.

Three components are required to construct the encoded circuit: initial state preparation $\mathcal{E}$, encoded unitary $\mathcal{U}$ and decoding $\mathcal{D}$. In our experiment, the first part will be a circuit that constructs a 4-qubit GHZ state as encoded $|00\rangle$ state, and the last part is a 4-qubit version of decoding circuit in Fig. 1. The second part circuit should implement non-Clifford logical operation $e^{-\mathrm{i}\theta Y_1X_2/2}$, which is encoded into $e^{-\mathrm{i}\theta Z_1X_2Y_4/2}$ according to eq. (2.1) and eq. (4.2). All above circuit blocks are combined to form a complete four-qubit physical circuit, as shown in the middle of Fig. 10. The bottom circuit of the same figure is a compiled version of physical circuit after mapping onto a linear configuration of qubits. Although this compilation may not be fully optimized, it would not affect the demonstration and performance of our protocol.

B. Experiment on Simulator

To demonstrate the protocol in Sec. II C, we simulate the compiled circuit in Fig. 10 using `qiskit` AerSimulator. We employ a 2-qubit depolarizing noise for all CNOT gates with error rate $p = 0.01$, which is consistent with the ECR error rate of selected qubits for the real device experiment (see Sec. IV C). We do not perform Pauli twirling in our simulation since the noise is already a Pauli channel.

There are two essential post-processing steps to perform the hybrid protocol: estimation of $\mathcal{N}_{\text{reduced}}$ and recovery of the noise-free expectation value from the data after post-selection. For the first step, we follow Sec. II D and firstly estimate the overall noise channel $\mathcal{N}_{\text{tot}}(\rho, \theta) = \sum_k p_kE_k(\theta)\rho E_k^\dagger(\theta)$. Here, E_k can be a single Pauli operator or a linear combination of Pauli operators, and θ is the parameter of the UCC ansatz. Then we set $p_k = 0$ if every term in $E_k(\theta)$ contains X or Y on any of the first two qubits, and renormalize the coefficients for $\mathcal{N}_{\text{logical}}$, which represents the remaining noise after applying QEDC. Finally, we discard the off-diagonal terms in

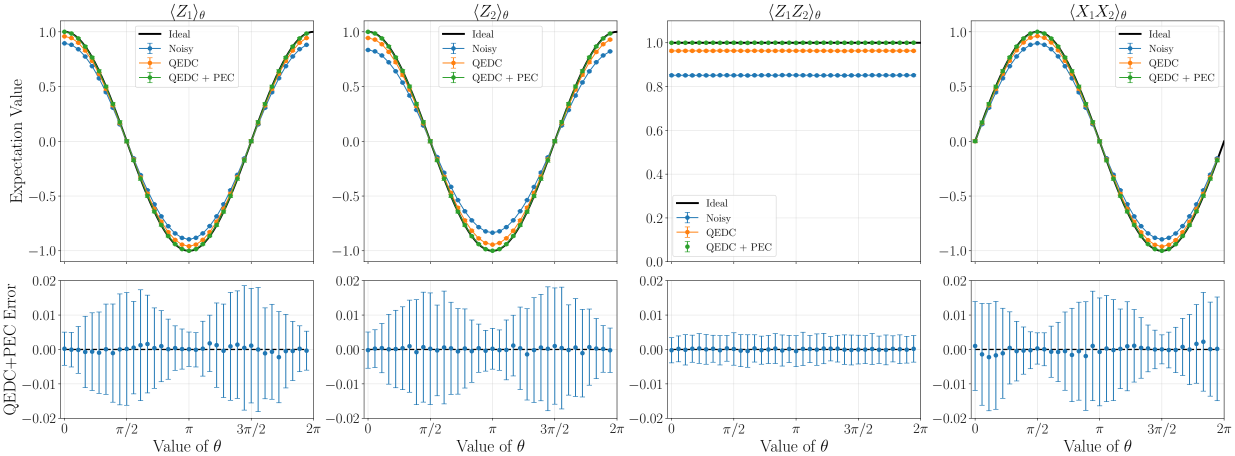

Figure 11. Simulation results of compiled circuit without any error suppression, with QEDC and with our combined protocol. The top panels show the averaged expectation values plus error bars at two standard deviation. The bottom panels give their differences from the exact values.

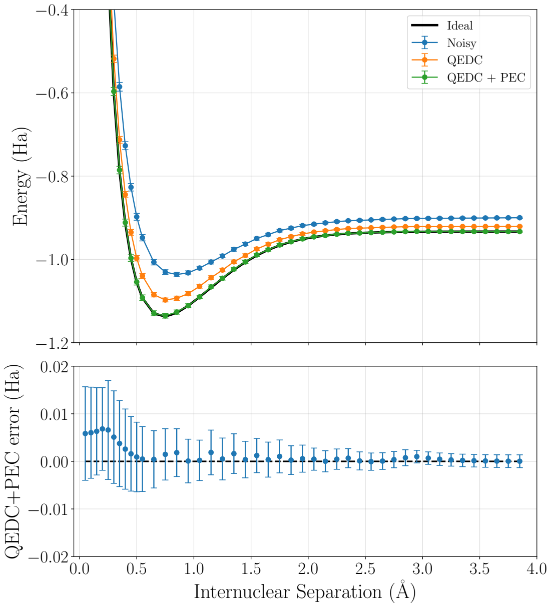

Figure 12. Simulation result of H_2 ground state energy as a function of internuclear distance. The top panel gives the result from different error suppression techniques, while the bottom panel only show the difference between exact value and the value from the protocol integrating QEDC and PEC.

$\mathcal{N}_{\text{logical}}$ and renormalize the coefficients again to obtain the reduced noise channel $\mathcal{N}_{\text{reduced}}$. In this case, $\mathcal{N}_{\text{reduced}}$ is a two-qubit Pauli channel and its inverse, $\mathcal{N}^{-1}_{\text{reduced}}(\theta) = \sum_{j=1}^{16} c_j^{\text{inv}}(\theta)\mathcal{P}_j$, can be computed from the inverse of Pauli transfer matrix [64] and the Walsh-Hadamard transformation.

For the recovery step, we estimate the noise-free expectation value via

$$\left\langle \widehat{A}(\theta) \right\rangle = \sum_{j=1}^{16} c_j^{\text{inv}}(\theta) \text{Tr}\left\{ A\,\mathcal{P}_j \circ \mathcal{N}_{\text{reduced}}[\rho(\theta)] \right\}, \tag{4.5}$$

where $\rho(\theta)$ is the two-qubit final state after decoding and post-selection. Note that in our paper, $\langle \widehat{A}(\theta) \rangle$ is not obtained from quasi-probability sampling; instead, we construct 16 compiled

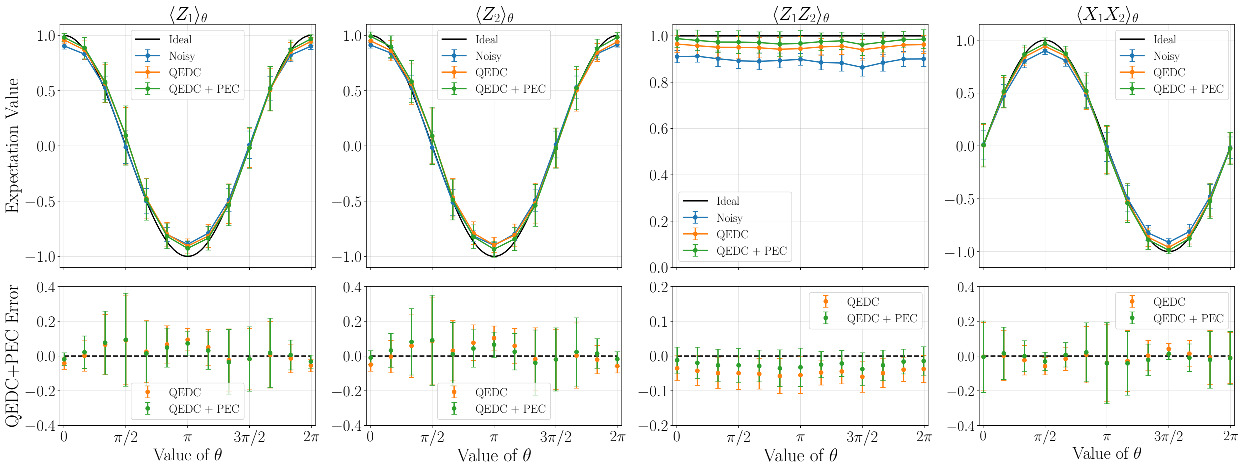

Figure 13. The expectation value obtained from `ibm_brussels`. All measurements are calculated after readout error mitigation. The result is consistent with the simulation result in Fig. 11.

circuits with different $P_1 \otimes P_2$ and sum over expectation values measured from each circuit after performing post-selection. We found that the final results show no significant difference between quasi-probability sampling and the direct summation.

Our simulation results are shown in Fig. 12, where each data point represents the average of expectation value from 100 independent simulations. Error bars indicate two standard deviations across experiments. Fig. 12 shows the potential energy surface of H_2 under different separation of two hydrogen atoms. Here, we adopt coefficients of Eq. (4.4) from Ref. [65] (See Appendix E). In order to compare the performance of different error suppression techniques, we also give the expectation values and the ground state energy obtained from data without any post-processing (labeled as "noisy") as well as data after selecting measurements with +1 for the first two qubits (labeled as "QEDC"). Our results indicate that the hybrid protocol can effectively recover the noisy expectation values to the ideal one when all physical noise is Pauli noise.

As mentioned in Sec. II D and Appendix B, the off-diagonal term in the χ-matrix of logical noise channel $\mathcal{N}_{\text{logical}}$ could affect the expectation value. We examine the logical noise channel in the simulation and conclude that under depolarizing error rate $p = 0.01$ for all CNOT gates, the second term in Eq. (2.21) will bring at most ± 0.001 bias to $\langle Z_1\rangle, \langle Z_2\rangle, \langle X_1X_2\rangle$ and $\pm 10^{-5}$ bias to $\langle Z_1Z_2\rangle$. This is negligible compared to the error bars of expectation values in Fig. 11.

C. Experiment on IBM Quantum Processors

We further demonstrate the integrated protocol using the VQE experiment together with partial Pauli twirling on the IBM Quantum `ibm_brussels` superconducting processor. First, we test compiled circuit without any error suppression on different configuration of qubits, and determine which of them is not significantly affected by hardware noise thus return reasonable result. As a consequence, we select a linear chain of qubits labeled as `19-18-14-0` to measure $\langle Z_1\rangle_\theta$, $\langle Z_2\rangle_\theta$, $\langle Z_1Z_2\rangle_\theta$, and $\langle X_1X_2\rangle_\theta$. We then perform cycle benchmarking [38, 66] to learn the Pauli fidelities $\boldsymbol{f}$ of ECR gates (native gate of `ibm_brussels`) acting on neighboring qubit pairs in the above two sets of qubit layout. The Pauli fidelities $\boldsymbol{f}$ should be converted into Pauli error rates $\boldsymbol{c}$ using the Walsh-Hadamard transformation for further usage (see Appendix A for details and results). Finally, we apply partial Pauli twirling on all ECR gates using twirling set $\{IY, IZ, YY, YZ, ZI, ZY, ZZ\}$, and run compiled circuits with light optimization (i.e., transpiler optimization level as 1) and other

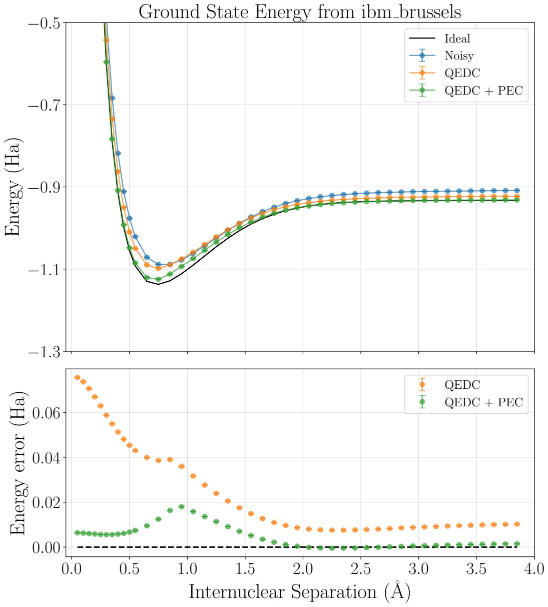

Figure 14. H_2 ground state energy estimated from real device experiments.

settings the same as those for simulation.

Fig. 13 plots the expectation values of the Hamiltonian under purely partial twirling, QEDC after partial twirling and the protocol integrating partial twirling, QEDC and PEC. We found that the readout error is a major source of noise that cannot be reduced by above error mitigation techniques. Hence, we employ the iterative Bayesian unfolding [19] with iteration as 2 to mitigate readout errors for all observables with minimum uncertainty. The error bars are estimated from the combination of variance of each term in Eq. (4.5), and are given at two standard deviations around the data points. Our results are consistent with the simulation results in Fig. 11, and the hybrid protocol can further suppress the noise and nearly recover noise-free expectation values as expected. The small discrepancy between experiment and ideal noise-free values may come from several sources, such as imperfect Pauli twirling, inaccurate noise estimation, and other hardware noise (single-qubit gate noise, crosstalk, etc.) that are not included in our noise model.

Finally, we interpolate the 13 data points in Fig. 13 with a cubic spline to obtain smooth expectation-value curves as a function of θ, and evaluate the potential energy surface of H_2 by minimizing Eq. (4.4) over θ at each fixed internuclear distance. Our results under different error reduction protocol are shown in Fig. 14. Here, the internuclear distance at which the energy is minimized under QEDC or the hybrid protocol matches that of the ideal curve (around 0.75Å), whereas the distance of minimum from the noisy data is slightly shifted to the right (between 0.75−0.85Å). Also, the energy minimum from the hybrid protocol is −1.124(7) Ha, which is very close to the exact value −1.137 Ha, while the minimum from noisy data and QEDC data are −1.088(8) Ha and −1.098(5) Ha, respectively. All these results suggest that the accuracy of VQE algorithm can be improved by our hybrid protocol. The main sources of bias in the energy estimate are the choice of interpolation method and parameters, as well as the limited number of data points.

To compare the sampling overhead of different approach, we compute γ^2 for overall error suppresion process using noise learning data from cycle benchmarking, and obtain $\gamma^2 \approx 1.19, 1.82, 1.86$ for our protocol, PEC at the end of circuit, and PEC after each layer, respectively. The results indicate that our protocol has lower sampling overhead than purely PEC, which is consistent with discussion in Sec. II E.

V. DISCUSSION AND CONCLUSION

While fault-tolerant quantum computing remains an long-term goal for the eventual success of quantum computation, a hybrid error suppression approach offers a promising path to near-term application. In this paper, we develop a hybrid protocol which integrates Pauli twirling and probabilistic error cancellation with the $[[n, n-2, 2]]$ quantum error detection code. The partial Pauli twirling scheme we proposed in Sec. III requires less than all 4^n Pauli operators. In contrast to Ref. [63] which also construct a smaller twirling set, our method is designed for the usage together with quantum codes since it manipulates noise channel such that it becomes more detectable or correctable. Moreover, the probabilistic error cancellation in Sec. II is modified to inverse noise after error detection instead of noise of each layer (see e.g., Ref. [17]), where the remaining noise is evaluated via Pauli error propagation. Recently, there have been some efforts in the literature that estimate and remove the overall noise of Clifford circuit/subcircuit [67, 68] with probabilistic error cancellation. However, these papers do not discuss how Pauli error propagates through single-qubit rotation gates and how to recover expectation under the impact of resulting non-stochastic noise, which is a crucial step in our protocol as described in Sec. II C-II E.

This framework offers a mutual advantage to quantum error detection code and quantum error mitigation. From the perspective of quantum error detection, both Pauli twirling and probabilistic error cancellation boost its performance because they eliminate undetectable errors in the encoded circuit. Meanwhile, numerical analysis in our paper shows that the sampling cost of probabilistic error cancellation is decreased when it is only used to remove undetectable errors in the encoded circuit. To demonstrate its utility on contemporary noisy hardware, we apply our protocol on a VQE circuit for H_2 ground state energy estimation and observe a significant improvement in the measured expectation value. This is the first quantum hardware implementation of a hybrid error suppression protocol based on PEC.

It is worth noting that even for a n-qubit Clifford circuits, probabilistic error cancellation for estimated overall noise is classically expensive, since it requires $O(4^n)$ classical memory to store complete information of n-qubit Pauli noise channel. To address this challenge, one should consider a sparse noise model for large quantum circuits that contains only the leading order errors. For example, a first-order noise model for individual n-qubit layer with k two-qubit gates includes $15k$ types of error if only one of the k gates is faulty, which is less resource demanding than the complete noise model. However, performing error mitigation for a sparse noise model may introduce more bias since second or higher order error may affect the result significantly when circuit size becomes sufficiently large. We leave a detailed analysis of the trade-off between efficiency and accuracy of using sparse noise model in future work.

During the preparation of this work, we become aware of an independent work that combines symmetry verification with probabilistic error cancellation to create the subspace noise tailoring algorithm [68]. Symmetry verification defines a computational subspace through a set of Pauli operators based on local fermion-to-qubit encoding, and it discards quantum noise that does not commute with any of these Pauli operators. It is obvious that symmetry verification acts similar with quantum error detection code, and probabilistic error cancellation in their protocol is also used to remove undetectable errors. However, the $[[n, n-2, 2]]$ quantum error detection codes always have two stabilizer generators and does not depend on the conserved quantities of problem Hamiltonian, so our protocol is easier to implement on a wider range of quantum algorithm, while their method may gives a better performance on fermionic simulation.

While most error mitigation techniques are designed to reduce bias in expectation value, it remains an open problem whether a hybrid approach could extend the capability of error mitigation beyond fixing noisy expectation values. This is possible because quantum error correction codes can protect arbitrary logical circuits, and probabilistic error cancellation is able to fully restore the measurement statistics in principle since it completely reverses the effect of the noise channel. However, all studies of hybrid error suppression, including our work, do not explore the potential to improve measurement statistics, even though some of these protocols are appropriate for such extension.

VI. ACKNOWLEDGMENTS

This work was supported by the U.S. Department of Energy (DOE) under Contract No. DE-AC02-05CH11231, through the Office of Advanced Scientific Computing Research Accelerated Research for Quantum Computing Program. This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract No. DE-AC05-00OR22725. This research was conducted in part using IBM Quantum Systems provided through USC's IBM Quantum Innovation Center. We thanks Meltem Tolunay, Nate Earnest-Noble and Siyuan Niu for helpful discussion on IBM real device implementation, and Ewout van den Berg for helpful discussion on noise learning and PEC based on Pauli-Lindblad noise model.

[1] J. Preskill, Quantum **2**, 79 (2018).
[2] A. Aspuru-Guzik *et al.*, Science **309**, 1704 (2005).
[3] S. Lloyd, Science **273**, 1073 (1996).
[4] P. W. Shor, SIAM Journal on Computing **26**, 1484 (1997).
[5] E. Farhi *et al.*, Science **292**, 472 (2001).
[6] P. W. Shor, Phys. Rev. A **52**, R2493 (1995).
[7] A. M. Steane, Phys. Rev. Lett. **77**, 793 (1996).
[8] A. R. Calderbank and P. W. Shor, Phys. Rev. A **54**, 1098 (1996).
[9] D. Gottesman, *Stabilizer codes and quantum error correction*, Ph.D. thesis, California Institute of Technology, Pasadena, CA (1997).
[10] Google Quantum AI, Nature **638**, 920 (2025).
[11] D. Bluvstein *et al.*, Nature **626**, 58 (2024).
[12] B. W. Reichardt *et al.*, "Demonstration of quantum computation and error correction with a tesseract code," (2024), arXiv:2409.04628 [quant-ph].
[13] C. Ryan-Anderson *et al.*, "High-fidelity and fault-tolerant teleportation of a logical qubit using transversal gates and lattice surgery on a trapped-ion quantum computer," (2024), arXiv:2404.16728 [quant-ph].
[14] P. S. Rodriguez *et al.*, "Experimental demonstration of logical magic state distillation," (2024), arXiv:2412.15165 [quant-ph].
[15] Y. Li and S. C. Benjamin, Phys. Rev. X **7**, 021050 (2017).
[16] K. Temme, S. Bravyi, and J. M. Gambetta, Phys. Rev. Lett. **119**, 180509 (2017).
[17] E. van den Berg *et al.*, Nature Physics **19**, 1116 (2023).
[18] J. J. Wallman and J. Emerson, Phys. Rev. A **94**, 052325 (2016).
[19] B. Nachman *et al.*, npj Quantum Information **6**, 84 (2020).
[20] Z. Cai *et al.*, Rev. Mod. Phys. **95**, 045005 (2023).
[21] Y. Kim *et al.*, Nature **618**, 500 (2023).
[22] T. E. O'Brien *et al.*, Nature Physics **19**, 1787 (2023).
[23] D. Aharonov and M. Ben-Or, SIAM Journal on Computing **38**, 1207 (2008).
[24] S. Bravyi and A. Kitaev, Phys. Rev. A **71**, 022316 (2005).
[25] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, 2010).
[26] R. Takagi *et al.*, npj Quantum Information **8**, 114 (2022).
[27] C. Piveteau *et al.*, Phys. Rev. Lett. **127**, 200505 (2021).
[28] M. Lostaglio and A. Ciani, Phys. Rev. Lett. **127**, 200506 (2021).
[29] Y. Suzuki *et al.*, PRX Quantum **3**, 010345 (2022).

[30] C. N. Self, M. Benedetti, and D. Amaro, Nature Physics **20**, 219 (2024).
[31] C. Gerhard and T. A. Brun, “Weakly fault-tolerant computation in a quantum error-detecting code,” (2024), arXiv:2408.14828 [quant-ph].
[32] M. Urbanek, B. Nachman, and W. A. de Jong, Phys. Rev. A **102**, 022427 (2020).
[33] K. Yamamoto *et al.*, Phys. Rev. Res. **6**, 013221 (2024).
[34] H. Nishi *et al.*, Phys. Rev. Appl. **23**, 034016 (2025).
[35] Z. He *et al.*, “Performance of quantum approximate optimization with quantum error detection,” (2025), arXiv:2409.12104 [quant-ph].
[36] A. G. Fowler, Phys. Rev. Lett. **109**, 180502 (2012).
[37] S. Ferracin *et al.*, Quantum **8**, 1410 (2024).
[38] S. Chen *et al.*, Nature Communications **14**, 52 (2023).
[39] C. Piveteau, D. Sutter, and S. Woerner, npj Quantum Information **8**, 12 (2022).
[40] D. Aharonov and M. Ben-Or, (1996), arXiv:quant-ph/9611025.
[41] A. R. Calderbank and P. W. Shor, Phys. Rev. A **54**, 1098 (1996), arXiv:quant-ph/9512032.
[42] E. Knill and R. Laflamme, Phys. Rev. Lett. **84**, 2525 (2000), arXiv:quant-ph/9604034.
[43] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Phys. Rev. A **86**, 032324 (2012), arXiv:1208.0928 [quant-ph].
[44] Y. Li and S. C. Benjamin, Phys. Rev. X **7**, 021050 (2017), arXiv:1611.09301 [quant-ph].
[45] J. R. McClean, M. E. Kimchi-Schwartz, J. Carter, and W. A. de Jong, Physical Review A **95** (2017), 10.1103/physreva.95.042308.
[46] K. Temme, S. Bravyi, and J. M. Gambetta, Phys. Rev. Lett. **119**, 180509 (2017), arXiv:1612.02058 [quant-ph].
[47] Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O’Brien, Rev. Mod. Phys. **95**, 045005 (2023), arXiv:2210.00921 [quant-ph].
[48] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters, Phys. Rev. Lett. **76**, 722 (1996).
[49] E. Knill, (2004), arXiv:quant-ph/0404104.
[50] O. Kern, G. Alber, and D. L. Shepelyansky, The European Physical Journal D **32**, 153–156 (2005).
[51] M. R. Geller and Z. Zhou, Physical Review A **88** (2013), 10.1103/physreva.88.012314.
[52] T. Scheiber, P. Haubenwallner, and M. Heller, (2024), arXiv:2412.01311 [quant-ph].
[53] A. Jain, P. Iyer, S. D. Bartlett, and J. Emerson, Physical Review Research **5** (2023), 10.1103/physrevresearch.5.033049.
[54] W. Munizzi, “Pauli-Twirling,” `https://github.com/WMunizzi/Pauli-Twirling`.
[55] L. Vaidman, L. Goldenberg, and S. Wiesner, Physical Review A **54**, R1745–R1748 (1996).
[56] E. M. Rains, (1997), arXiv:quant-ph/9704043.
[57] P. Iyer, A. Jain, S. D. Bartlett, and J. Emerson, Physical Review Research **4** (2022), 10.1103/physrevresearch.4.043218.
[58] C. J. Wood, J. D. Biamonte, and D. G. Cory, Quant. Inf. Comput. **15**, 0759 (2015), arXiv:1111.6950 [quant-ph].
[59] J. Guillaud, J. Cohen, and M. Mirrahimi, SciPost Physics Lecture Notes **72** (2023), 10.21468/SciPostPhysLectNotes.72.
[60] A. A. U. of Chicago et al., npj Quantum Information (2023), 10.1038/s41534-023-00746-0.
[61] S. et al., PRX Quantum **5** (2024), 10.1103/PRXQuantum.5.020355.
[62] Q. Xu, H. Putterman, J. Iverson, K. Noh, and O. Painter, *Quantum Computing, Communication, and Simulation II; Proceedings of SPIE*, (2022).
[63] Z. Cai and S. C. Benjamin, Scientific Reports **9** (2019), 10.1038/s41598-019-46722-7.
[64] D. Greenbaum, “Introduction to quantum gate set tomography,” (2015), arXiv:1509.02921 [quant-ph].
[65] J. I. Colless *et al.*, Phys. Rev. X **8**, 011021 (2018).
[66] A. Erhard *et al.*, Nature Communications **10**, 5347 (2019).
[67] T. Scheiber, P. Haubenwallner, and M. Heller, “Reducing pec overhead by pauli error propagation,” (2025), arXiv:2412.01311 [quant-ph].
[68] M. Papič, M. G. Algaba, E. Godinez-Ramirez, I. de Vega, A. Auer, F. Šimkovic IV, and A. Calzona, “Near-term fermionic simulation with subspace noise tailored quantum error mitigation,” (2025), arXiv:2503.11785 [quant-ph].
[69] S. Endo, S. C. Benjamin, and Y. Li, Phys. Rev. X **8**, 031027 (2018).

Appendix A: Learning the Pauli Noise via Cycle Benchmarking

A n-qubits Pauli noise channel is defined as $\Lambda(\rho) = \sum_j c_j \mathcal{P}_j(\rho)$ where the Pauli error rates $\boldsymbol{c} = \{c_j\}$ forms a normal probability distribution. The Pauli channel can be also written in the form of Pauli transfer matrix $\mathcal{T}$ [64], which is defined as

$$\mathcal{T}_{ab} = \frac{1}{2^n} \mathrm{Tr}\left[P_a \Lambda(P_b)\right]. \tag{A1}$$

The Pauli transfer matrix for a n-qubit Pauli channel has only diagonal elements. These elements are called Pauli fidelities $\boldsymbol{f} = \{f_k\}$, where the k-th element is defined by $f_k = \mathrm{Tr}\left[P_k \Lambda(P_k)\right]/2^n$. The transformation of Pauli fidelities $\boldsymbol{f}$ and Pauli error rates $\boldsymbol{c}$ is given by $\boldsymbol{f} = \boldsymbol{W}\boldsymbol{p}$, or

$$f_k = \sum_j (-1)^{\langle P_j, P_k \rangle} c_j,\ c_j = \frac{1}{4^n} \sum_k (-1)^{\langle P_j, P_k \rangle} f_k. \tag{A2}$$

Here $\langle P_j, P_k \rangle$ denotes the symplectic inner product of Pauli operators, which is zero if two operators commute or one if they anti-commute. Such a transformation is referred to as Walsh-Hadamard transformation in the literature.

The Walsh-Hadamard transform reveals an important property of a n-qubit Pauli noise channel Λ: each of 4^n Pauli operators P_k is an eigen-operator of channel Λ with eigenvalue f_k (so it is also called Pauli eigenvalues). This can be seen from following equation,

$$\Lambda(P_k) = \sum_j c_j P_j P_k P_j = \sum_j (-1)^{\langle P_j, P_k \rangle} c_j P_k = f_k P_k, \tag{A3}$$

Cycle benchmarking and its variants can be used to efficiently characterize Pauli channel information [66]. To be specific, one can directly measure the Pauli fidelities $\boldsymbol{f}$ with cycle benchmarking, then derive Pauli error rates $\boldsymbol{c}$ from Walsh-Hadamard transformation. To understand the process of learning Pauli fidelities, here we provide an example of learning individual f_{ZI} of ECR gate noise using cycle benchmarking circuit in Fig. S1. At the beginning, the circuit prepares the +1 eigenstate of operator ZI with both $\mathcal{R}_1$ and $\mathcal{R}_2$ as identity. If tracing the state with its stabilizer, the first set of random Pauli gates $\mathcal{P}_{11}$ and $\mathcal{P}_{12}$ will add a phase factor of ±1 to the stabilizer. It will still be ZI after propagating through the first ECR gate, then becomes $f_{ZI} \cdot ZI$ according to eq. (A3) after applying the noise channel (we ignore the ±1 phase here). After four noisy ECR gates the stabilizer becomes $(f_{ZI})^4 ZI$. At the end of the circuit, we measure operator ZI and compute $\mathrm{Tr}[ZI \cdot \widetilde{\mathcal{U}}(\rho)] = \mathrm{Tr}[ZI \cdot (f_{ZI}^4 ZI\rho)] = A_{ZI} f_{ZI}^4$, where $\widetilde{\mathcal{U}}$ denotes the circuit with four noisy ECR gates and five sets of random Pauli. The coefficient A_{ZI} is related to state preparation and measurement error, but it is independent with depth of cycle benchmarking circuit. Finally, one repeat above experiment with m repetition of ECR for f_{ZI}^m, and perform an exponential fitting $\mathrm{CB}(m) = A f_{ZI}^m$ to get SPAM-error free Pauli fidelity f_{ZI}.

However, some Pauli fidelities cannot be learned from cycle benchmarking due to the existence of degeneracy [17, 38]. The issue of unlearnability can be seen from below example. Suppose we are going to learn the Pauli fidelity f_{XI} with top circuit in Fig. S1. The circuit prepares the +1 eigenstate of XI with $\mathcal{R}_1$ as Hadamard gate. Then stabilizer XI will change to YX after the first ECR gate, and become $f_{YX} \cdot YX$ after passing through the noise channel. The second ideal ECR will convert YX back to XI, and the noise channel of the second ECR add a coefficient f_{XI} to the stabilizer, which gives $f_{XI} f_{YX} XY$. Repeating this process one eventually obtain the stabilizer $(f_{XI} f_{YX})^2 XY$, which lead to $A_{XI,YX}(f_{XI} f_{YX})^2$ as the expectation value of XI. Here, the Pauli fidelity f_{XI} cannot be isolated from aforementioned measurement result, so we say the Pauli fidelity f_{XI} is unlearnable.

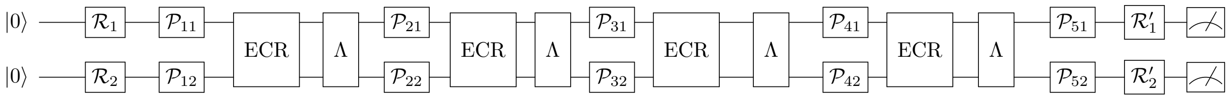

Figure S1. Cycle benchmarking circuit with four ECRs. Here $\mathcal{R}_{1,2}$ and $\mathcal{R}'_{1,2}$ rotate $|00\rangle$ state to appropriate basis. $\mathcal{P}_{j1}$ and $\mathcal{P}_{j2}$ are random Pauli gates used for Pauli twirling on the first and second qubits, respectively.

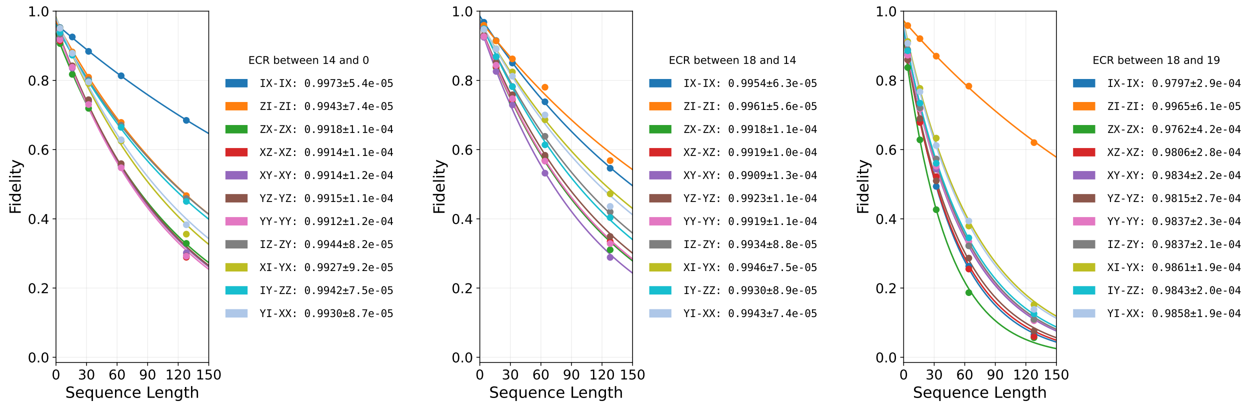

Figure S2. The exponential fitting and results of Pauli fidelities for ECR gate on qubit pair with label `14-0`, `18-14` and `18-19` in `ibm_brussels`. If the result is purely for one Pauli fidelity, e.g., f_{IX}, we label the results as "IX-IX". Here, the qubit pair shows the default "direction" of ECR gates in the given device. If there is degeneracy, e.g., $\sqrt{f_{XI}f_{YX}}$, we label the results as "XI-YX". This result is used for the QEDC+PEC protocol in the real device VQE experiment.

Following the discussion in Ref. [38], the Pauli fidelity of IX, XY, XZ, YY, YZ, ZI and ZX of Pauli noise channel of ECR gate are learnable via cycle benchmarking, while the degeneracy issue of $f_{IZ}f_{ZY}$, $f_{XI}f_{YX}$, $f_{IY}f_{ZZ}$ and $f_{YI}f_{XX}$ cannot be solved by any quantum experiment, even using interleaved gates $\sqrt{Z}$ and $\sqrt{X}$ [38] before each ECR gates. An intuition of the unlearnability is that one cannot recover identity operator in the stabilizer before or after ECRs. In this paper, we assume the Pauli fidelity of each unlearnable pair is equal, even though they can be any possible value (e.g., αf_{IZ} and f_{ZY}/α) as long as their product is equal to the measurement.

In our real device demonstration, we run the cycle benchmarking experiment on a linear configuration of qubits with label as `19-18-14-0` on the `ibm_brussels`. During the experiment, all ECRs are assumed to have identical noise model if they act on same controlled-target qubit pair. Following open-source code from Ref. [38], we build up cycle benchmarking circuit for basis of following nine operators, $\{XX, XY, XZ, YX, YY, YZ, ZX, ZY, ZZ\}$, on each qubit pair with ECR sequences length as $4, 16, 32, 64, 128$. We execute individual circuit under optimization level as 1 on the IBM device and compute $\mathrm{Tr}\left[P_k \widetilde{\mathcal{U}}(\rho)\right]$ for each of nine Pauli operators. For those basis IX and XI, we use the result from circuits preparing the basis of XX to compute the expectation value, since IX and XI are also stabilizer of $+1$ eigenstate of XX. Similarly, we measure the expectation of IY from result of circuit preparing XY, IZ from XZ, YI from YX and ZI from ZX. Eventually, we perform an exponential fitting $\mathrm{CB}(m) = Af_k^m$ for each Pauli basis, where m is the length of CNOT sequences, and obtain a SPAM-free Pauli fidelity f_k. The experimental results are shown in Fig. S2. Note that f_k is already free from readout error so it is not necessary to apply the measurement error mitigation on benchmarking results. The Pauli error rate $\boldsymbol{c}$ is then derived from the Pauli fidelities $\boldsymbol{f}$ using the Walsh-Hadamard transformation, which is not shown here but used for the noise estimation in VQE experiment discussed in Sec. IV C.

Appendix B: Error Propagation with Rotation

Consider the error propagation identity $\mathcal{V}\circ\mathcal{P}=\mathcal{M}\circ\mathcal{V}$ shown in eq. (2.15), where $\mathcal{V}(\rho)=V\rho V^\dagger$ and $\mathcal{M}(\rho)=M\rho M^\dagger$. If V is a rotation gate, then M can be derived by following relations. For $R_x(\theta)$ gate,

$$\begin{aligned} R_x(\theta)XR_x^\dagger(\theta) =&X, &\text{(B1)}\\ R_x(\theta)YR_x^\dagger(\theta) =&\cos\theta Y+\sin\theta Z, &\text{(B2)}\\ R_x(\theta)ZR_x^\dagger(\theta) =&\cos\theta Z-\sin\theta Y. &\text{(B3)}\end{aligned}$$

For $R_y(\theta)$ gate,

$$\begin{aligned} R_y(\theta)XR_y^\dagger(\theta) =&\cos\theta X-\sin\theta Z, &\text{(B4)}\\ R_y(\theta)YR_y^\dagger(\theta) =&Y, &\text{(B5)}\\ R_y(\theta)ZR_y^\dagger(\theta) =&\cos\theta Z+\sin\theta X. &\text{(B6)}\end{aligned}$$

For $R_z(\theta)$ gate,

$$\begin{aligned} R_z(\theta)XR_z^\dagger(\theta) =&\cos\theta X+\sin\theta Y,\\ R_z(\theta)YR_z^\dagger(\theta) =&\cos\theta Y-\sin\theta X,\\ R_z(\theta)ZR_z^\dagger(\theta) =&Z. \end{aligned} \tag{B7}$$

If M is a linear combination of Pauli operator, the channel $\mathcal{M}$ has non-zero off-diagonal terms in its χ-matrix representation, $\mathcal{M}(\rho)=\sum_{j,k}\chi_{jk}P_j\rho P_k$, so $\mathcal{M}$ is no longer a Pauli channel. This can be seen from an example of X error after propagating through $R_z(\theta)$ gate, where the error becomes $M=\cos\theta X+\sin\theta Y$ and the channel below is not a Pauli channel,

$$\begin{aligned}\mathcal{M}=&(\cos\theta X+\sin\theta Y)\rho(\cos\theta X+\sin\theta Y)^\dagger\\ =&\cos^2\theta X\rho X+\cos\theta\sin\theta(X\rho Y+Y\rho X)+\sin^2\theta Y\rho Y.\end{aligned} \tag{B8}$$

According to the discussion of eq. (2.4), the $[[n,n-2,2]]$ code is able to reduce noise in the form of $\mathcal{N}(\rho)=\sum_s q_sE_s\rho E_s$, where $E_s=\sum_s a_{r_s}P_s$ could be a linear combination of Pauli and $\{a_{r_s}^2\}$ gives a probability distribution. For example, the noise in Eq. (B8) can be fully removed by the QEDC since both X and Y anti-commute with its Z-type stabilizer generator. However, in general, QEDC cannot eliminate all possible errors. In such cases, the noisy expectation value is given by

$$\mathrm{Tr}\left[A\,\mathcal{N}_{\text{logical}}(\rho)\right]=\sum_{s,r_s}q_sa_{r_s}^2\,\mathrm{Tr}\left[A\,P_{r_s}\rho P_{r_s}\right]+\sum_{\substack{s,r_s,k_s\\ r_s\neq k_s}}q_sa_{r_s}a_{k_s}\,\mathrm{Tr}\left[A\,P_{r_s}\rho P_{k_s}\right]. \tag{B9}$$

It is resource consuming to completely recover above expectation value via PEC, which will require 16^n basis operations to implement the inverse of a n-qubit logical error channel [69]. Instead, one can approximate the noiseless expectation value by neglecting the impact from off-diagonal terms and cancelling only the diagonal part of the error channel, i.e., $\mathcal{N}_{\text{reduced}}(\rho)=\sum_s q_sa_{r_s}^2P_{r_s}\rho P_{r_s}$, which only requires at most 4^n basis operations to implement the Pauli channel $\mathcal{N}_{\text{reduced}}^{-1}(\rho)$.

It is worth mentioning that the contribution of some off-diagonal terms to the noisy expectation value can be neglected. We can rewrite each term in the second summation of eq. (B9) as follows,

$$\begin{aligned}\mathrm{Tr}\left[A\,P_{r_s}\rho P_{k_s}\right]=&\mathrm{Tr}\left[P_{k_s}A\,P_{r_s}\rho\right]\\ =&\mathrm{Tr}\left[(-1)^{\langle P_{k_s},A\rangle}(-1)^{\langle P_{k_s},P_{r_s}\rangle}(-1)^{\langle P_{r_s},A\rangle}P_{r_s}A\,P_{k_s}\rho\right]\\ =&\mathrm{Tr}\left[(-1)^{\langle P_{k_s},A\rangle}(-1)^{\langle P_{k_s},P_{r_s}\rangle}(-1)^{\langle P_{r_s},A\rangle}A\,P_{k_s}\rho P_{r_s}\right]\end{aligned} \tag{B10}$$

where $\langle P_A, P_B\rangle$ is sympletic inner product of two Pauli operators P_A and P_B. In this context, the off-diagonal terms $\mathrm{Tr}\left[A\,P_{r_s}\rho P_{k_s}\right]$ and $\mathrm{Tr}\left[A\,P_{k_s}\rho P_{r_s}\right]$ in eq. (B9) will cancel if $P_{k_s}A\,P_{r_s}$ anticommutes with $P_{r_s}A\,P_{k_s}$. Moreover, the noisy expectation may not be significantly affected by remaining off-diagonal part when the error rate q_s is very small. Therefore, we believe that inversing $\mathcal{N}_{\text{reduced}}$ will possibly gives a good approximation of ideal expectation value.

For the purpose of illustration, we explain how the idea of canceling $\mathcal{N}_{\text{reduced}}$ works on a single-qubit example. Suppose $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ is a noiseless single-qubit state, then the ideal expectation value of observable Z is given by $\langle\psi|Z|\psi\rangle = |\alpha|^2 - |\beta|^2$. When an error $M = \cos\theta X + \sin\theta Y$ occurs, according to Eq. (B8), the noisy expectation value is derived by

$$\begin{aligned}\mathrm{Tr}\left[Z\,\mathcal{M}(\rho)\right] =& \cos^2\theta \mathrm{Tr}\left[ZX\rho X\right] + \cos\theta\sin\theta \mathrm{Tr}\left[ZX\rho Y\right] \\ &+ \cos\theta\sin\theta \mathrm{Tr}\left[ZY\rho X\right] + \sin^2\theta \mathrm{Tr}\left[ZY\rho Y\right] \\ =& -|\alpha|^2 + |\beta|^2 + \cos\theta\sin\theta \mathrm{Tr}\left[ZX\rho Y\right] + \cos\theta\sin\theta \mathrm{Tr}\left[ZY\rho X\right] \\ =& -|\alpha|^2 + |\beta|^2.\end{aligned} \tag{B11}$$

Here, $XZY = -YZX$ so the off-diagonal terms in the second equality vanish, and the noisy expectation value is equivalent with the one under $\mathcal{N}_{\text{reduced}}(\rho) = \cos^2\theta X\rho X + \sin^2\theta Y\rho Y$, where

$$\mathrm{Tr}\left[Z\,\mathcal{N}_{\text{reduced}}(\rho)\right] = \cos^2\theta \mathrm{Tr}\left[ZX\rho X\right] + \sin^2\theta \mathrm{Tr}\left[ZY\rho Y\right] = -|\alpha|^2 + |\beta|^2. \tag{B12}$$

In this example, the ideal expectation value can be recovered by implementing $\mathcal{N}^{-1}_{\text{reduced}}(\rho)$ via quasi-probability sampling.

Appendix C: Sampling Overhead of our Protocol

In our toy model for evaluation and comparison, we consider n-qubits depolarizing noise [25] for each layer,

$$\mathcal{N}(\rho) = (1-p)\rho + p\frac{I}{2^n}, \tag{C1}$$

where $I/2^n$ is the completely mixed state. The Pauli transfer matrix of above channel is a $4^n \times 4^n$ diagonal matrix given by

$$R_{\mathcal{N}} = \mathrm{diag}(1, 1-p, \ldots, 1-p). \tag{C2}$$

Here we consider three settings of error mitigation in the numerical test. The first protocol is regular PEC protocol that cancel noise after each layer. The second protocol is PEC that cancel estimated overall noise. Although the idea of cancelling overall noise is proposed in Sec. II D for our combined protocol, it can be still applied independently. The third protocol is the one proposed in Sec. II.

We then compute the sampling overhead for each protocol on a $n = 4$ qubits circuit without any unitary operation (see Fig. S3). For the noise estimation in the second and third protocol, we use a property that the Pauli transfer matrix of channel $\mathcal{N}_2 \circ \mathcal{N}_1$ is the product of Pauli transfer matrices of the two channels, i.e., $R_{\mathcal{N}_2\circ\mathcal{N}_1} = R_{\mathcal{N}_2}R_{\mathcal{N}_1}$ [64]. For the last protocol, we check and remove all Pauli terms in total noise that anticommute with stabilizer generators $X^{\otimes n}$ and $Z^{\otimes n}$ for post-selection, and renormalized the coefficient to obtain $\mathcal{N}_{\text{reduced}}$. Note that all noise channels, including $\mathcal{N}_{\text{tot}}$ and $\mathcal{N}_{\text{reduced}}$, are Pauli channels, and their inverse, $\mathcal{N}^{-1} = \sum_j c_j^{\text{inv}} P_j\rho P_j$, can be computed by Walsh-Hadamard transform [17]

$$c_j^{\text{inv}} = \frac{1}{4^n}\sum_k (-1)^{\langle P_j, P_k\rangle}\frac{1}{f_k}, \tag{C3}$$

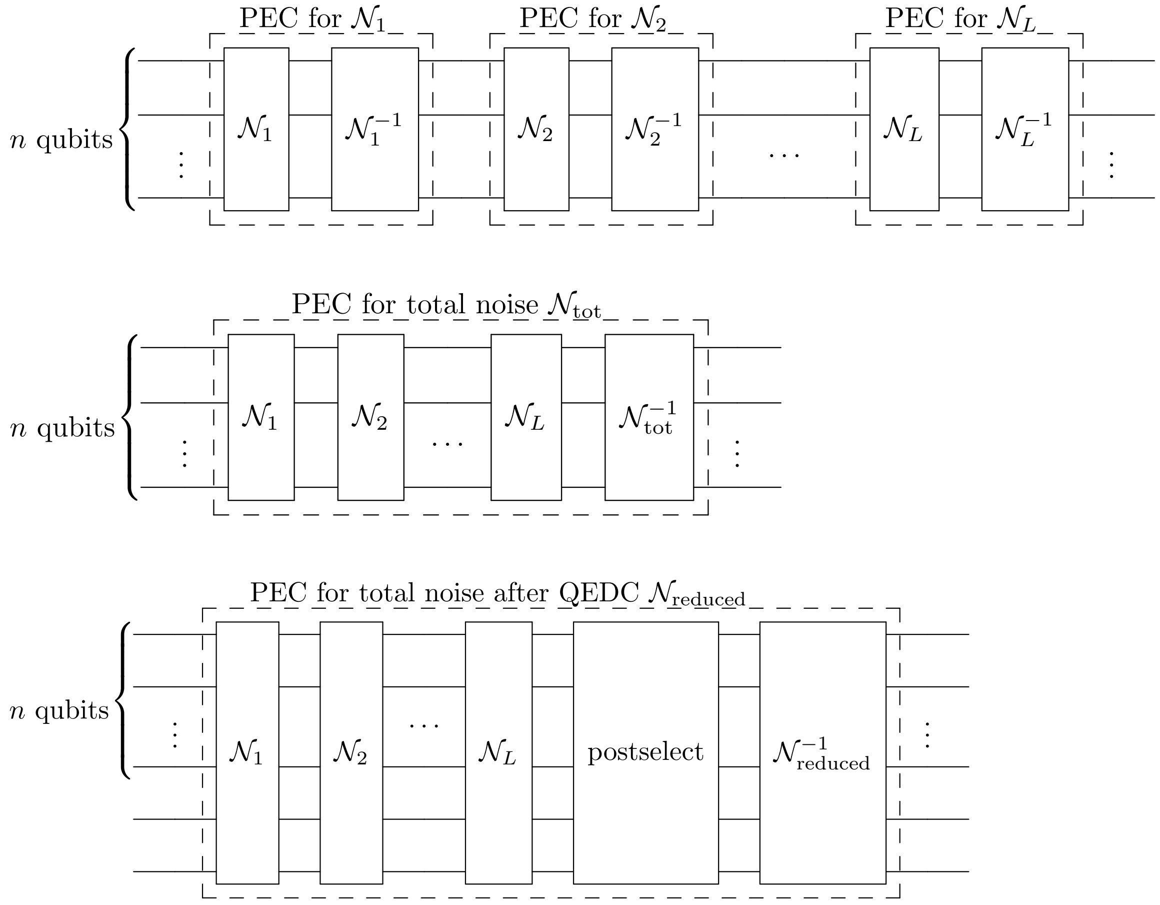

Figure S3. Three protocols for overhead analysis. The top one is for the regular PEC. The middle one is for PEC canceling overall noise. The bottom one is for our combined protocol with QEDC and PEC.

where f_k is the Pauli fidelity of noise channel $\mathcal{N}$. Finally, one would have $\gamma = \sum_j |c_j^{\text{inv}}|$ for each inverse channel, and γ^2 for sampling overhead.

Appendix D: Additional Cycle Benchmarks

In this section we provide additional cycle benchmarking results for native multi-qubit unitaries on the IBM quantum platform. In case case we apply no twirling, Pauli twirling, and partial Pauli twirling, in combination with a 3-qubit bit-flip code as defined in Section III.

For the CZ gate, we benchmark the noisy unitary $\tilde{\mathcal{U}} = CZ_{1,2}CZ_{2,3}$.

For the ECR gate, we benchmark the noisy unitary $\tilde{\mathcal{U}} = ECR1, 2ECR2, 3$.

For the $iSWAP$ gate, we benchmark the noisy unitary $\tilde{\mathcal{U}} = iSWAP_{1,2}iSWAP_{2,3}$.

Appendix E: Coefficients for Hamiltonian

Table I shows the coefficients for Eq. (4.4) as a function of the internuclear separation R from Ref. [65]. The electronic integrals are calculated using in the STO-3G (GTO) basis set.

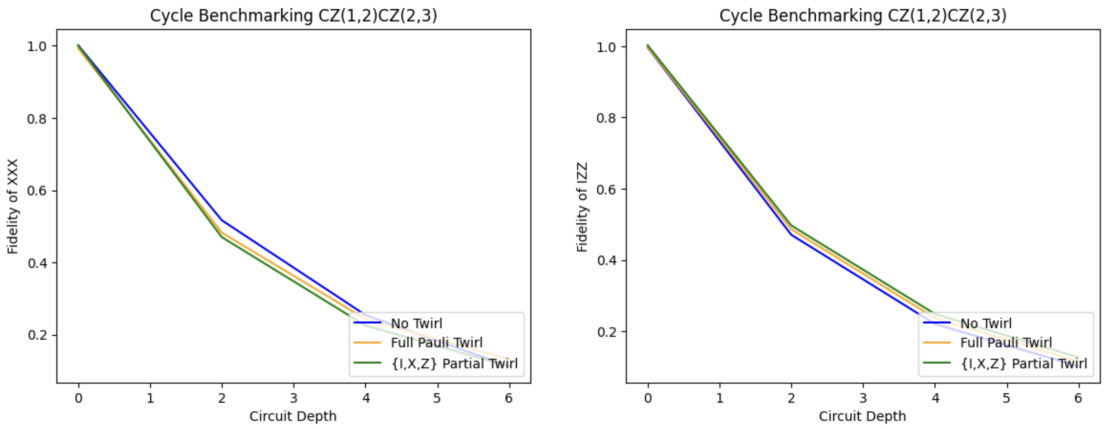

Figure S4. Cycle benchmarking performed on $\tilde{\mathcal{U}} = CZ_{1,2}CZ_{2,3}$ using twirling with bit-flip error correction. Experiment conducted on ibm brussels device. The fidelity of GHZ_3 stabilizers XXX and IZZ are plotted against increasing twirled layers of $\tilde{\mathcal{U}}$. For partial twirling, the twirling gates are selected to maximize X error propagation, and minimize all others. Performances compared between no twirling, full Pauli twirling using I, X, Y, Z gates, and partial twirling using I, X, Z gates.

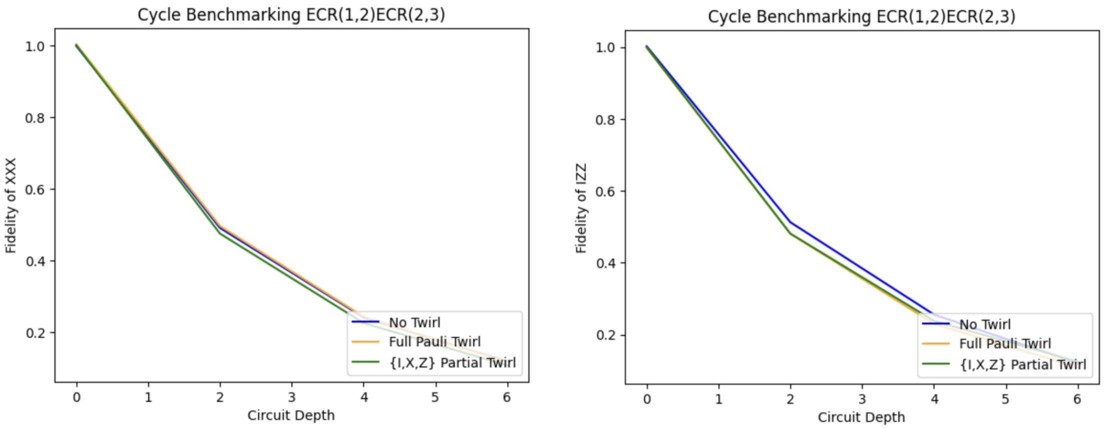

Figure S5. Cycle benchmarking performed on $\tilde{\mathcal{U}} = ECR_{1,2}ECR_{2,3}$ using twirling with bit-flip error correction. Experiment conducted on ibm brussels device. The fidelity of GHZ_3 stabilizers XXX and IZZ are plotted against increasing twirled layers of $\tilde{\mathcal{U}}$. For partial twirling, the twirling gates are selected to maximize X error propagation, and minimize all others. Performances compared between no twirling, full Pauli twirling using I, X, Y, Z gates, and partial twirling using I, X, Z gates.

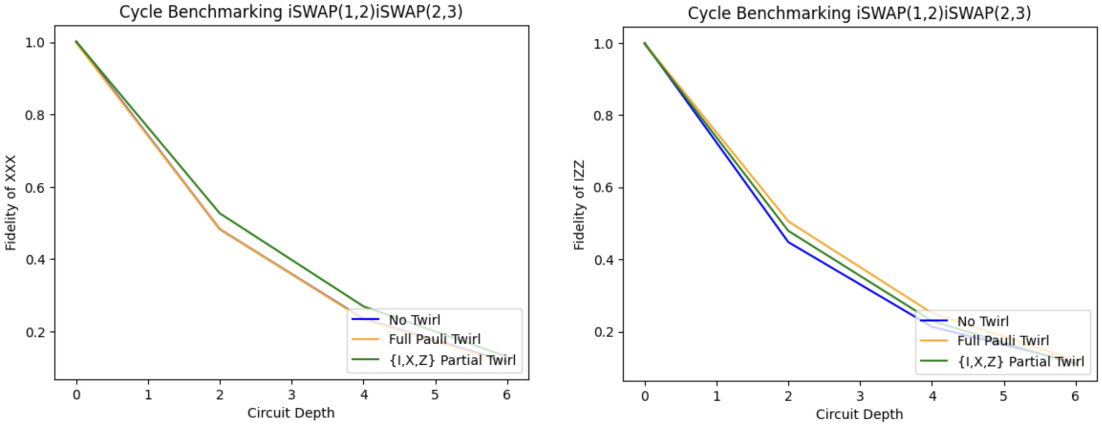

Figure S6. Cycle benchmarking performed on $\tilde{\mathcal{U}} = iSWAP_{1,2}iSWAP_{2,3}$ using twirling with bit-flip error correction. Experiment conducted on ibm brussels device. The fidelity of GHZ_3 stabilizers XXX and IZZ are plotted against increasing twirled layers of $\tilde{\mathcal{U}}$. For partial twirling, the twirling gates are selected to maximize X error propagation, and minimize all others. Performances compared between no twirling, full Pauli twirling using I, X, Y, Z gates, and partial twirling using I, X, Z gates.

Table I.

R (Å)	g_1	g_2	g_5	g_3	g_4
0.05	10.0777	-1.05533	0.155708	-1.05533	0.0139333
0.1	4.75665	-1.02731	0.15617	-1.02731	0.0138667
0.15	2.94817	-0.984234	0.15693	-0.984234	0.013761
0.2	2.01153	-0.930489	0.157973	-0.930489	0.0136238
0.25	1.42283	-0.870646	0.159277	-0.870646	0.0134635
0.3	1.01018	-0.808649	0.160818	-0.808649	0.013288
0.35	0.701273	-0.747416	0.162573	-0.747416	0.0131036
0.4	0.460364	-0.688819	0.164515	-0.688819	0.012914
0.45	0.267547	-0.63389	0.166621	-0.63389	0.0127192
0.5	0.110647	-0.58308	0.16887	-0.58308	0.0125165
0.55	-0.0183734	-0.536489	0.171244	-0.536489	0.0123003
0.65	-0.213932	-0.455433	0.176318	-0.455433	0.0118019
0.75	-0.349833	-0.388748	0.181771	-0.388748	0.0111772
0.85	-0.445424	-0.333747	0.187562	-0.333747	0.0104061
0.95	-0.513548	-0.287796	0.19365	-0.287796	0.00950345
1.05	-0.5626	-0.248783	0.199984	-0.248783	0.00850998
1.15	-0.597973	-0.215234	0.206495	-0.215234	0.00747722
1.25	-0.623223	-0.186173	0.213102	-0.186173	0.00645563
1.35	-0.640837	-0.160926	0.219727	-0.160926	0.00548623
1.45	-0.652661	-0.138977	0.226294	-0.138977	0.0045976
1.55	-0.660117	-0.119894	0.23274	-0.119894	0.00380558
1.65	-0.664309	-0.103305	0.239014	-0.103305	0.00311545
1.75	-0.666092	-0.0888906	0.245075	-0.0888906	0.0025248
1.85	-0.666126	-0.0763712	0.250896	-0.0763712	0.00202647
1.95	-0.664916	-0.0655065	0.256458	-0.0655065	0.001611
2.05	-0.662844	-0.0560866	0.26175	-0.0560866	0.00126812
2.15	-0.660199	-0.0479275	0.266768	-0.0479275	0.000988
2.25	-0.657196	-0.0408672	0.271512	-0.0408672	0.000761425
2.35	-0.653992	-0.0347636	0.275986	-0.0347636	0.000580225
2.45	-0.650702	-0.0294924	0.280199	-0.0294924	0.000436875
2.55	-0.647408	-0.0249459	0.28416	-0.0249459	0.000325025
2.65	-0.644165	-0.0210309	0.287881	-0.0210309	0.0002388
2.75	-0.641011	-0.0176672	0.291376	-0.0176672	0.0001733
2.85	-0.637971	-0.0147853	0.294658	-0.0147853	0.0001242
2.95	-0.635058	-0.0123246	0.297741	-0.0123246	8.7875e-05
3.05	-0.632279	-0.0102318	0.300638	-0.0102317	6.145e-05
3.15	-0.629635	-0.00845958	0.303362	-0.00845958	4.2425e-05
3.25	-0.627126	-0.00696585	0.305927	-0.00696585	2.895e-05
3.35	-0.624746	-0.0057128	0.308344	-0.0057128	1.955e-05
3.45	-0.622491	-0.0046667	0.310625	-0.0046667	1.305e-05
3.55	-0.620353	-0.00379743	0.31278	-0.00379743	8.575e-06
3.65	-0.618325	-0.0030784	0.314819	-0.0030784	5.6e-06
3.75	-0.616401	-0.00248625	0.31675	-0.00248625	3.6e-06
3.85	-0.614575	-0.00200063	0.318581	-0.00200062	2.275e-06
3.95	-0.612846	-0.00160393	0.32032	-0.00160392	1.425e-06