Supplementary Information for "Demonstrating quantum error mitigation on logical qubits"

Aosai Zhang,[1,*] Haipeng Xie,[2,*] Yu Gao,[1,*] Jia-Nan Yang,[1] Zehang Bao,[1] Zitian Zhu,[1] Jiachen Chen,[1] Ning Wang,[1] Chuanyu Zhang,[1] Jiarun Zhong,[1] Shibo Xu,[1] Ke Wang,[1] Yaozu Wu,[1] Feitong Jin,[1] Xuhao Zhu,[1] Yiren Zou,[1] Ziqi Tan,[1] Zhengyi Cui,[1] Fanhao Shen,[1] Tingting Li,[1] Yihang Han,[1] Yiyang He,[1] Gongyu Liu,[1] Jiayuan Shen,[1] Han Wang,[1] Yanzhe Wang,[1] Hang Dong,[1] Jinfeng Deng,[1,3] Hekang Li,[1,4] Zhen Wang,[1,4] Chao Song,[1,4] Qiujiang Guo,[1,4] Pengfei Zhang,[1,4,†] Ying Li,[2,‡] and H. Wang[1,3,4]

[1]*School of Physics, ZJU-Hangzhou Global Scientific and Technological Innovation Center, and Zhejiang Key Laboratory of Micro-nano Quantum Chips and Quantum Control, Zhejiang University, Hangzhou, China*
[2]*Graduate School of China Academy of Engineering Physics, Beijing 100193, China*
[3]*State Key Laboratory of Extreme Photonics and Instrumentation, College of Optical Science and Engineering, Zhejiang University, Hangzhou 310027, China*
[4]*Hefei National Laboratory, Hefei 230088, China*

CONTENTS

Supplementary Note 1. ZERO-NOISE EXTRAPOLATION FORMULAS

First, we introduce some notations and briefly review NISQ and FTQC circuits. Here, FTQC circuits refer to quantum circuits with mid-circuit state preparation and measurement, feedback and post-selection. Then, we show how to express an FTQC circuit with an example. Finally, we generalize the expression to arbitrary FTQC circuits. With the expression, we derive the formula for ZNE.

Notations. We use X_i, Y_i, Z_i to denote Pauli operators on qubit-i, and we use $\mathbb{1}$ to denote the identity operator. Given an operator V, we use $[V]$ to denote a completely positive map $[V]\bullet = V \bullet V^\dagger$.

We use three types of primitive operations to implement quantum computing: gate, state preparation and measurement. We can denote them with completely positive maps. An ideal gate is denoted by $[U]$, where U is a unitary operator. An ideal state preparation is denoted by $\mathcal{A}_{P,i} = [(\mathbb{1} + P_i)/2] + [P_i'][(\mathbb{1} - P_i)/2]$, which prepares qubit-$i$ in the eigenstate of $P = X, Y, Z$ with the eigenvalue $+1$. Here, P' is an arbitrary Pauli operator different from P, and $(\mathbb{1} \pm P_i)/2$ is the projection operator onto eigenstates of P_i with the eigenvalue ± 1. An ideal measurement is denoted by $\mathcal{B}_{P,i}(\mu) = [(\mathbb{1} + \mu P_i)/2]$, which is a measurement on qubit-i in the $P = X, Y, Z$ basis. Here, $\mu = \pm 1$ is the measurement outcome. The maps $[U]$ and $\mathcal{A}_{P,i}$ are trace-preserving; $\mathcal{B}_{P,i}(\mu)$ is not, but $\mathcal{B}_{P,i}(+1) + \mathcal{B}_{P,i,}(-1)$ is trace-preserving.

When errors occur stochastically in an operation, the actual operation is in the form $\mathcal{M} = (1-p)\mathcal{M}^I + p\mathcal{M}^E$, where $\mathcal{M}^I = [U], \mathcal{A}_{P,i}, \mathcal{B}_{P,i}(\mu)$ is the ideal operation, p is the probability of errors, and $\mathcal{M}^E$ is a completely positive map denoting the operation with errors.

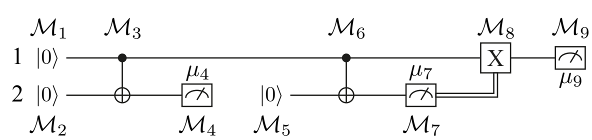
Now, we take the Pauli error model as an example, which is a practical mode of errors in quantum computing. In the Pauli error model, the process generating Pauli errors is a map in the form $\mathcal{N} = (1-p)[\mathbb{1}] + p\mathcal{E}$, where p is the error probability, and $\mathcal{E}$ is a trace-preserving completely positive map denoting errors. If we neglect crosstalk, errors only happen on the qubit subset that an operation acts on, called the support. For a single-qubit operation on qubit-i, the noise map is $\mathcal{N}_1 = (1-p)[\mathbb{1}] + (p_X[X_i] + p_Y[Y_i] + p_Z[Z_i])$, where P_X, P_Y, P_Z are probabilities of corresponding Pauli errors, and $p = p_X + p_Y + p_Z$ is the total error probability. Similarly, for a two-qubit gate on qubits i and j, the noise map is $\mathcal{N}_2 = (1-p)[\mathbb{1}] + \sum_{P=X,Y,Z}(p_{PI}[P_i] + p_{IP}[P_j]) + \sum_{P,P'=X,Y,Z} p_{PP'}[P_iP'_j]$, where $p = \sum_{P=X,Y,Z}(p_{PI} + p_{IP}) + \sum_{P,P'=X,Y,Z} p_{PP'}$. To include crosstalk, the general noise map is in the form $\mathcal{N} = (1-p)[\mathbb{1}] + \sum_P p_P[P]$, where P is a Pauli operator, p_P is the probability of the Pauli error P, and $p = \sum_P p_P$. Here, the Pauli operator P is either a single-qubit Pauli operator or an arbitrary product of single-qubit operators, and it may act on qubits out of operation's support. In this general Pauli-error noise map, the erroneous map is $\mathcal{E} = p^{-1}(\sum_P p_P[P])$.

In the Pauli error model, the actual operation of a gate is in the form $\mathcal{M} = \mathcal{N}\mathcal{M}^I$, where $\mathcal{M}^I = [U]$ is the ideal gate. For a state preparation operation, the actual operation is also in the form $\mathcal{M} = \mathcal{N}\mathcal{M}^I$, where $\mathcal{M}^I = \mathcal{A}_{P,i}$ is the ideal state preparation. For a measurement, the actual operation is in the form $\mathcal{M}(\mu) = \mathcal{M}^I(\mu)\mathcal{N}$ (the noise occur before the ideal operation this time), where $\mathcal{M}^I(\mu) = \mathcal{B}_{P,i}(\mu)$ is the ideal measurement. Substituting the expression of the noise map $\mathcal{N}$, we can find that actual operations are in the form $\mathcal{M} = (1-p)\mathcal{M}^I + p\mathcal{M}^E$, where $\mathcal{M}^E = \mathcal{E}\mathcal{M}^I$ for gates and state preparation, and $\mathcal{M}^E = \mathcal{M}^I\mathcal{E}$ for measurement.

NISQ circuits and FTQC circuits. In an NISQ circuit, qubits are prepared at the beginning and measured at the end of the circuit, and gates are applied between the state preparation and measurement. Such circuits are insufficient for the usual protocols of FTQC.

In error correction, we usually detect errors using measurements. Typically, we repeatedly measure a set of carefully chosen Pauli operators, stabilizer operators of the error correction code. Then, we send the measurement outcomes to a classical algorithm called a decoder. With the algorithm, we compute a set of Pauli gates. By applying the Pauli gates, we can correct errors on qubits. Therefore, the error correction is a typical feedback process.

We need post-selection when the magic state is used in quantum computing. Because of the error correction code, we cannot directly apply all necessary gates on logical qubits. The magic state represents a promising protocol to complete the universal gate set. In the magic-state protocol, we prepare some resource states on logical qubits and then improve their fidelities in certain distillation circuits. With the resource states distilled, we can use them to implement gates that cannot be implemented directly. The distillation circuits improve the fidelity through post-selection: the resource state is kept or discarded depending on measurements in the distillation circuit; if discarded, we need to re-prepare the state and try distillation for another round. We remark that the preparation, distillation and utilization of magic states also require feedback operations.



Supplementary Fig. 1. An example circuit with feedback and post-selection operations. The circuit succeeds when $\mu_4 = +1$.

An example FTQC circuit. To illustrate the expression of FTQC circuits using completely positive maps, we take the circuit in Supplementary Fig. 1 as an example, which includes feedback and post-selection operations. There are three state preparation operations and three measurements in the circuit. Depending on the measurement outcome μ_7, the number of gates is either two or three.

Using completely positive maps, we can express the ideal circuit in the following form

$$\begin{aligned} &\mathcal{M}^I(\mu_4,\mu_7,\mu_9) \\ &= \delta_{\mu_4,+1}\mathcal{M}^I_9(\mu_9)\mathcal{M}^I_8(\mu_7)\mathcal{M}^I_7(\mu_7)\mathcal{M}^I_6 \\ &\quad\times\mathcal{M}^I_5\mathcal{M}^I_4(\mu_4)\mathcal{M}^I_3\mathcal{M}^I_2\mathcal{M}^I_1. \end{aligned} \tag{1}$$

Here, $\mathcal{M}^I_1 = \mathcal{A}_{Z,1}$ and $\mathcal{M}^I_2 = \mathcal{M}^I_5 = \mathcal{A}_{Z,2}$ are state preparation operations, $\mathcal{M}^I_3 = \mathcal{M}^I_6 = [U]$ are controlled-NOT gates, $\mathcal{M}^I_4(\mu_4) = \mathcal{B}_{Z,2}(\mu_4)$, $\mathcal{M}^I_7(\mu_7) = \mathcal{B}_{Z,2}(\mu_7)$ and $\mathcal{M}^I_9(\mu_9) = \mathcal{B}_{Z,1}(\mu_9)$ are measurements, $\mathcal{M}^I_8(\mu_7) = \delta_{\mu_7,+1}[\mathbb{1}] + \delta_{\mu_7,-1}[X_1]$ is the measurement-dependent gate in the feedback operation, and $\delta_{\mu_4,+1}$ denotes the post-selection operation.

With noise, the map of the circuit becomes

$$\begin{aligned}&\mathcal{M}(\mu_4,\mu_7,\mu_9)\\&=\delta_{\mu_4,+1}\mathcal{M}_9(\mu_9)\mathcal{M}_8(\mu_7)\mathcal{M}_7(\mu_7)\mathcal{M}_6\\&\times\mathcal{M}_5\mathcal{M}_4(\mu_4)\mathcal{M}_3\mathcal{M}_2\mathcal{M}_1,\end{aligned} \tag{2}$$

where operations are in the from $\mathcal{M}_j=(1-p_j)\mathcal{M}_j^I+p_j\mathcal{M}_j^E$. Because the eighth operation is measurement-dependent, it has a measurement-dependent error probability $p_8(\mu_7)$ and erroneous operation $\mathcal{M}_8^E(\mu_7)$: $p_8(+1)$ and $\mathcal{M}_8^E(+1)$ [$p_8(-1)$ and $\mathcal{M}_8^E(-1)$] are the error probability and erroneous operation, respectively, of the gate $\mathbb{1}$ (X_1).

General FTQC circuits. For a general FTQC circuit, we can express the ideal circuit consisting of N operations in the form

$$\begin{aligned}\mathcal{M}^I(\tilde{\mu}_N)=g(\tilde{\mu}_N)&\mathcal{M}_N^I(\tilde{\mu}_{N-1},\mu_N)\cdots\\&\times\mathcal{M}_j^I(\tilde{\mu}_{j-1},\mu_j)\cdots\mathcal{M}_2^I(\tilde{\mu}_1,\mu_2)\mathcal{M}_1^I(\mu_1).\end{aligned} \tag{3}$$

Here, $\mathcal{M}_j^I$ is the completely positive map denoting the jth operations. We use μ_j to represent the measurement outcome of the jth operation: $\mu_j=\pm1$ ($\mu_j=0$) if the jth operation is (not) a measurement. If the jth operation is a measurement, the map $\mathcal{M}_j^I$ is a function of the measurement outcome μ_j; even if the jth operation is not a measurement, we still can express it as a function of μ_j because μ_j takes only one value anyway. We use $\tilde{\mu}_j$ to represent a tuple of measurement outcomes from the first to jth operations, and we can define it formally with the recursion formula $\tilde{\mu}_j=(\tilde{\mu}_{j-1},\mu_j)$ and the initial value $\tilde{\mu}_0=()$. The jth operation may depend on measurement outcomes of previous operations since the feedback. Therefore, the map $\mathcal{M}_j^I$ is also a function of $\tilde{\mu}_{j-1}$: If the operation is not a feedback operation, the dependence is trivial, i.e. $\mathcal{M}_j^I(\tilde{\mu}_{j-1},\mu_j)=\mathcal{M}_j^I(\mu_j)$, such as operations with $j\neq 8$ in the example circuit; the eighth operation is the only one with a non-trivial dependence on previous measurement outcomes. The function g describes the post-selection,

$$g(\tilde{\mu}_N)=\begin{cases}1, & \tilde{\mu}_N\in S;\\0, & \tilde{\mu}_N\notin S,\end{cases} \tag{4}$$

where S denotes the set of measurement outcomes indicating success.

For the noisy circuit, its map is in the form

$$\begin{aligned}\mathcal{M}(\tilde{\mu}_N)=g(\tilde{\mu}_N)&\mathcal{M}_N(\tilde{\mu}_{N-1},\mu_N)\cdots\\&\times\mathcal{M}_j(\tilde{\mu}_{j-1},\mu_j)\cdots\mathcal{M}_2(\tilde{\mu}_1,\mu_2)\mathcal{M}_1(\mu_1),\end{aligned} \tag{5}$$

where operations are in the from

$$\begin{aligned}\mathcal{M}_j(\tilde{\mu}_{j-1},\mu_j)=[1-p_j(\tilde{\mu}_{j-1})]&\mathcal{M}_j^I(\tilde{\mu}_{j-1},\mu_j)\\&+p_j(\tilde{\mu}_{j-1})\mathcal{M}_j^E(\tilde{\mu}_{j-1},\mu_j).\end{aligned} \tag{6}$$

If the jth operation is a feedback operation that depends on previous measurement outcomes, such as the eighth operation in the example circuit, error probability p_j depends on $\tilde{\mu}_{j-1}$; otherwise, p_j is a constant function; the erroneous operation $\mathcal{M}_j^E$ is similar.

The expression in Supplementary Eq. (5) illustrates the temporal order of operations that an operation can only depend on measurement outcomes in previous operations. This temporal order is unimportant for error mitigation, and if neglect, we have a simplified expression of a noisy circuit: We use $\boldsymbol{\mu}=\tilde{\mu}_N$ to denote all measurement outcomes in the circuit, then

$$\mathcal{M}(\boldsymbol{\mu})=g(\boldsymbol{\mu})\mathcal{M}_N(\boldsymbol{\mu})\cdots\mathcal{M}_j(\boldsymbol{\mu})\cdots\mathcal{M}_2(\boldsymbol{\mu})\mathcal{M}_1(\boldsymbol{\mu}), \tag{7}$$

where operations are in the from

$$\mathcal{M}_j(\boldsymbol{\mu})=[1-p_j(\boldsymbol{\mu})]\mathcal{M}_j^I(\boldsymbol{\mu})+p_j(\boldsymbol{\mu})\mathcal{M}_j^E(\boldsymbol{\mu}). \tag{8}$$

Expansion formula of the observable with errors. With a circuit, we can evaluate observables that are functions of measurement outcomes, $f(\boldsymbol{\mu})$. For example, if we want to evaluate the observable Z_1 in the example circuit, we take $f(\boldsymbol{\mu})=\mu_9$. In the case of error correction, this function takes into account the last-round error correction on the measurement outcomes.

At this point, we discuss the role of the decoding algorithm, which maps error syndromes to corresponding correction operations. These operations are performed at designated points within the circuit. Suppose the jth operation is a correction step. The associated error syndrome, denoted by a tuple $\mathbf{s}$, is determined from the outcomes of previous parity-check measurements, i.e., $\mathbf{s} = S(\tilde{\mu}_{j-1})$, where S is a function of the prior measurement outcomes $\tilde{\mu}_{j-1}$. The correction operation is typically a Pauli operator defined by $\sigma = E(\mathbf{s}) = E(S(\tilde{\mu}_{j-1}))$, where E represents the decoding algorithm. Thus, the jth operation can be expressed as $\mathcal{M}_j = [E(S(\tilde{\mu}_{j-1}))]$. In addition to Pauli-gate corrections, it is often necessary to correct errors in data-qubit measurements by flipping the measurement outcomes. Let $\mathbf{s}' = S'(\boldsymbol{\mu})$ denote the relevant error syndrome, derived from the measurement outcomes $\boldsymbol{\mu}$. The decoding algorithm outputs a tuple of signs $\boldsymbol{\theta} = E'(\mathbf{s}) = E'(S'(\boldsymbol{\mu}))$, where a negative sign indicates a flip of the corresponding outcome. If $f_0(\boldsymbol{\mu})$ is the observable function in an error-free circuit, then the corrected observable function becomes $f(\boldsymbol{\mu}) = f_0(\boldsymbol{\theta\mu})$, where $\boldsymbol{\theta\mu}$ denotes the entry-wise product, and $\boldsymbol{\theta}$ is itself a function of $\boldsymbol{\mu}$. In summary, the effects of the decoding algorithm are fully incorporated into the circuit operations $\mathcal{M}_j(\boldsymbol{\mu})$ and the final observable function $f(\boldsymbol{\mu})$.

The ideal expected value of the observable is

$$\langle O\rangle_{\mathrm{ideal}} = \sum_{\boldsymbol{\mu}} f(\boldsymbol{\mu}) \operatorname{Tr}\left[\mathcal{M}^I(\boldsymbol{\mu})\rho\right], \tag{9}$$

where ρ is the initial state, and $\operatorname{Tr}\left[\mathcal{M}^I(\boldsymbol{\mu})\rho\right]$ is the probability of the measurement outcome $\boldsymbol{\mu}$ in the ideal circuit.

Because of the post-selection, the distribution may not be normalized in general. The expected value in the normalized distribution is

$$\langle O\rangle_{\mathrm{ideal}} = \frac{\langle O\rangle_{\mathrm{ideal}}}{\langle \mathbb{1}\rangle_{\mathrm{ideal}}}, \tag{10}$$

and we have the expression of $\langle \mathbb{1}\rangle_{\mathrm{ideal}}$ by substituting $f = 1$ into Supplementary Eq. (9). In the following, we focus on $\langle O\rangle$, and the result can be applied to $\langle \mathbb{1}\rangle$.

With a noisy circuit, the expected value of the observable is

$$\langle O\rangle = \sum_{\boldsymbol{\mu}} f(\boldsymbol{\mu}) \operatorname{Tr}\left[\mathcal{M}(\boldsymbol{\mu})\rho\right], \tag{11}$$

where $\operatorname{Tr}\left[\mathcal{M}(\boldsymbol{\mu})\rho\right]$ is the probability of the measurement outcome $\boldsymbol{\mu}$ in the noisy circuit. To explicitly express the observable as a function of error probabilities, we rewrite each actual operation in the circuit as

$$\mathcal{M}_j = \mathcal{M}_j^{(0)} + p_j \mathcal{M}_j^{(1)} = \sum_{b_j=0,1} p_j^{b_j} \mathcal{M}_j^{(b_j)}, \tag{12}$$

where $\mathcal{M}_j^{(0)} = \mathcal{M}_j^I$ is the ideal operation, and

$$\mathcal{M}_j^{(1)} = \mathcal{M}_j^E - \mathcal{M}_j^I \tag{13}$$

is the deviation from the ideal operation. Substituting this expression of $\mathcal{M}_j$, we have

$$\begin{aligned}\langle O\rangle = &\sum_{\boldsymbol{\mu}} \sum_{b_1,b_2,\ldots,b_N=0,1} \left[\prod_{j=1}^{N} p_j^{b_j}(\boldsymbol{\mu})\right] \\ &\times a(b_1, b_2, \ldots, b_N; \boldsymbol{\mu}),\end{aligned} \tag{14}$$

where

$$\begin{aligned}&a(b_1, b_2, \ldots, b_N; \boldsymbol{\mu}) \\ &= f(\boldsymbol{\mu}) g(\boldsymbol{\mu}) \operatorname{Tr}\left[\mathcal{M}_N^{(b_N)}(\boldsymbol{\mu}) \cdots \mathcal{M}_1^{(b_1)}(\boldsymbol{\mu})\rho\right].\end{aligned} \tag{15}$$

Eventually, we can rewrite the expression as

$$\langle O\rangle = \langle O\rangle_{\mathrm{ideal}} + \sum_{k=1}^{N} a_k, \tag{16}$$

where

$$a_k = \sum_{\boldsymbol{\mu}} \sum_{j_1<j_2<\cdots<j_k} p_{j_1}(\boldsymbol{\mu})p_{j_2}(\boldsymbol{\mu})\cdots p_{j_k}(\boldsymbol{\mu})a(j_1, j_2, \ldots, j_k; \boldsymbol{\mu}), \tag{17}$$

and

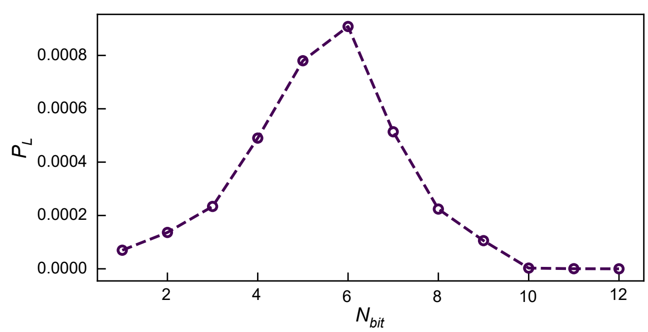
$$\begin{aligned} &a(j_1, j_2, \ldots, j_k) \\ &= a(b_1, b_2, \ldots, b_N)|_{b_j=1 \text{ iff } j\in\{j_1,j_2,\ldots,j_N\}}. \end{aligned} \tag{18}$$

Here, we have used that $\sum_{\boldsymbol{\mu}} a(0, 0, \ldots, 0; \boldsymbol{\mu}) = \langle O\rangle_{\text{ideal}}$.

In ZNE, we mitigate errors by boosting the error probability. If we boost errors in each operation with the same factor of r (i.e. $p_j \to rp_j$ for all j), the computing result becomes a polynomial function of r, $\langle O\rangle(r) = \langle O\rangle_{\text{ideal}} + \sum_{k=1}^{N} a_k r^k$. This formula holds for both NISQ and FTQC circuits.

Supplementary Note 2. CORRELATIONS IN LOGICAL ERRORS OF QLDPC CODES

Unlike the surface code, many qLDPC codes encode a large number of logical qubits in a single block of physical qubits. Errors in these logical qubits could be significantly correlated, tending to occur simultaneously on multiple logical qubits, as illustrated in Supplementary Fig. 2. In the $[[72, 12, 6]]$ code [1], the probability that 4 to 7 logical qubits fail simultaneously is higher than other cases, indicating significant correlations. For such correlated errors, measuring their error rates is extremely challenging. Even assuming Pauli errors, obtaining a model of correlated errors requires measuring $4^k - 1$ error rates, where k is the number of logical qubits in a block.



Supplementary Fig. 2. **Numerical results of multi-qubit correlations in a bivariate bicycle code.** We take the $[[72, 12, 6]]$ code in the bivariate bicycle family and a physical error rate of $p = 0.002$ per operation as an example. Using Monte Carlo simulation, we evaluate the number of logical qubits N_{bit} that are affected by logical errors simultaneously in each trial. The probability of each qubit number is ploted.

Supplementary Note 3. GENERATING CIRCUIT INSTANCES AND CALCULATING THE EXPECTATIONS

To controllably amplify existing errors, we choose to insert an operation drawn from the list of $\{I, X, Y, Z\}$ during the operational stage for each qubit. The most intuitive way to generate adequate circuit instances for the estimation of $\langle Z_{\text{L}}\rangle$ (or $\langle X_{\text{L}}\rangle$) — randomly selecting the operation with assigned probabilities for each circuit instance — can be inefficient. For example, in the scenario of using a distance-3 (7) repetition code, around 80% (60%) of circuit instances are devoid of any error injection when $p = 3.6\%$ and $r = 1$. In addition, circuit instances with the presence of multiple injected errors, which can potentially lead to the failure of quantum error correction and contribute to the infidelity of estimation, are unlikely to be chosen. Alternatively, we can iterate over all possible circuit instances and numerically construct the desired circuit list by calculating the weighted average. For example, in the experiment related to Fig. 2 of the main text, all 4^3 circuit instances were experimentally implemented for each value of r, and the

Supplementary Table 1. **Qubit parameters, coherence properties, measurement fidelities, and gate performance for Processor I and Processor II.**

	Processor I			Processor II		
Parameter	Median	Mean	Stdev.	Median	Mean	Stdev.
Qubit idle frequency, $\omega_{\mathrm{idle}}/2\pi$ (GHz)	4.000	3.962	0.128	4.123	4.120	0.131
Qubit anharmonicity, $\alpha/2\pi$ (MHz)	-197.3	-197.1	1.6	-214.6	-214.7	2.3
Readout frequency, $\omega_{\mathrm{r}}/2\pi$ (GHz)	6.316	6.299	0.095	6.416	6.416	0.086
Energy relaxation time, T_1 (μs)	124.36	111.82	33.22	127.89	132.44	24.32
Spin-echo dephasing time, T_2^{SE} (μs)	11.34	11.72	2.35	15.55	18.18	7.81
Readout error e_{r} using Method I / Method II (%)	0.47 / 0.87	0.52 / 0.82	0.16 / 0.18	0.87 / -	0.94 / -	0.30 / -
1Q XEB Pauli error, e_1 (%)	0.085	0.079	0.021	0.055	0.052	0.019
2Q XEB Pauli error, e_2 (%)	0.56	0.56	0.14	0.37	0.40	0.17

resulting data were processed to estimate the expectations of Z_0. However, in repetition and surface codes, the large number of circuit instances makes full implementation impractical. To address this challenge, we employ a method to select the most likely circuit instances as follows.

First, we determine the total number of circuit instances, which scales proportionally with the code distance to ensure more accurate estimations (see Supplementary Table 2). Second, we consider the case of injecting k errors into the circuit and calculate the required number of circuit instances based on the error-injection probability rp. Circuit instances are then randomly selected without replacement until the desired number is reached or all possible instances have been selected. The number of remaining circuit instances to be determined is updated accordingly. Finally, the second and third steps are iteratively repeated, starting with $k = 0$ and continuing until the total number is reached.

The above procedure assumes perfect operations and measurements. In practice, however, when the number of possible circuit instances for injecting k errors is small — such as in the case of $k = 0$ — the statistical error due to finite measurement shots can proportionately impact the overall estimation accuracy, scaling with the assigned weight. To reduce this effect, we introduce a key modification to the procedure: the circuit number for each k is required to exceed a predefined threshold, such as 1% of the total circuit number. Equivalently, this modification increases the number of measurement shots, thereby reducing statistical errors and improving the accuracy of the estimation.

Supplementary Table 2. **Number of circuit instances used to estimate the expectation values.**

	1-round	2-round	3-round	4-round
distance-3 repetition code	1000	1000	1500	2000
distance-5 repetition code	3500	5000	5000	6000
distance-7 repetition code	6000	6000	6000	7000
distance-3 surface code	4000	-	-	-

We estimate the expectation values, such as $\langle Z_{\mathrm{L}} \rangle$ and $\langle X_{\mathrm{L}} \rangle$, by averaging measured results across all generated circuit instances with the respective weights. Specifically, we denote $O_{k,c,s}$ as the corresponding outcome with k errors are injected into the circuit, with c representing the circuit index and s the index of measurement shot. The expectation value is then calculated as

$$\overline{O} = \frac{1}{S} \sum_{k,c,s} P(k) \frac{O_{k,c,s}}{C(k)}, \tag{19}$$

where S represents the number of measurement shots for each circuit instance, $C(k)$ denotes the number of chosen circuits with k errors injected, and $P(k)$ is the respective weight. The value of S for experiments of repetition code and surface code is 150.

Supplementary Note 4. MORE EXPERIMENTAL DETAILS

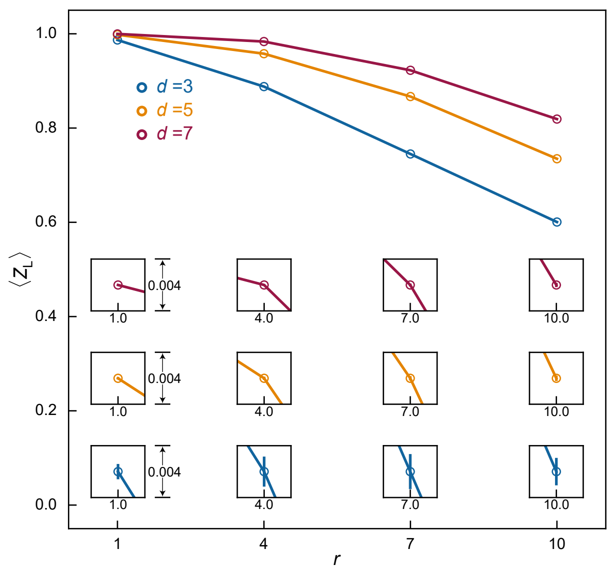
Standard error of the measured expectations. As mentioned in Supplementary Note 3, S measurement shots are taken for each circuit. It allows us to directly calculate the standard deviation of $\overline{O}$ (see Supplementary Eq. 19) without requiring additional experimental effort. First, we calculate the average expectation value for each shot index s as

$$\overline{O}_s = \sum_{k,c} P(k) \frac{O_{k,c,s}}{C(k)}, \tag{20}$$

where the sum is taken over the error count k and circuit instance c. Next, the unbiased standard deviation of $\overline{O}$ can be obtained by

$$\sigma\left[\overline{O}\right] = \sqrt{\frac{1}{S(S-1)} \sum_s \left(\overline{O}_s - \overline{O}\right)^2}. \tag{21}$$

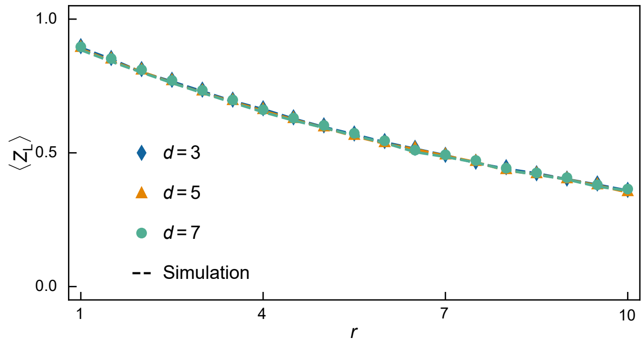
We note that the standard errors are relatively small for both the repetition code and the surface code, making them difficult to visualize clearly in the main figures. See Supplementary Fig. 3 for an enlarged view of the standard errors for the one-round repetition code.



Supplementary Fig. 3. **Standard error of the measured $\langle Z_{\mathrm{L}} \rangle$.** All insets have a spread of 0.004 along the y-axis, with the data points positioned at the center of the figures.

Uncorrected expectation values for the repetition code. In Supplementary Fig. 4, we observe that the uncorrected expectation values for different repetition code distances nearly overlap. The slight differences observed for large noise scaling factors are mainly due to the randomness of the circuit instances, which also influence the simulation results. These uncorrected values incorporate errors from single-qubit gates, two-qubit gates, and readout operations — whether occurring during logical state preparation or parity checks. This ensures that they provide a fair and reliable baseline for comparison with the corrected ones.

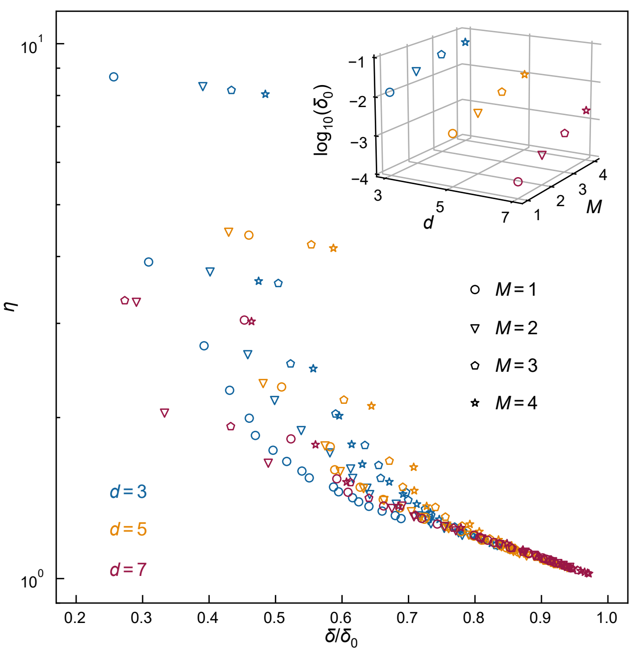
Investigating the scalability of ZNE with a fixed error rate per parity-check round. Unlike in the main text, where the unit error probability for each M is adjusted to maintain a fixed total error rate (see Supplementary Table 3), in this study, the unit error probability in each parity-check round is fixed at $p = 3.6\%$ and then the performance of ZNE is evaluated. As depicted in Supplementary Fig. 5, the relative bias, represented by δ/δ_0, consistently remains below 1, confirming the effectiveness of ZNE. The ZNE results demonstrate consistent trends for different code distances and parity-check round numbers, with a larger code distance resulting in a smaller sampling overhead. Additionally, the sampling overhead remains nearly unchanged for different numbers of parity-check rounds. These findings demonstrate the scalability of ZNE for error correction circuits, even with increasing circuit complexity.



Supplementary Fig. 4. **Uncorrected results of the one-round repetition code.** For different repetition code distances, the uncorrected expectation values of Z_{L} remain nearly the same. Dashed lines: noisy numerical simulations.

Supplementary Table 3. **Unit error probabilities.** For each M, the unit error probability is chosen so that the total error rate is fixed.

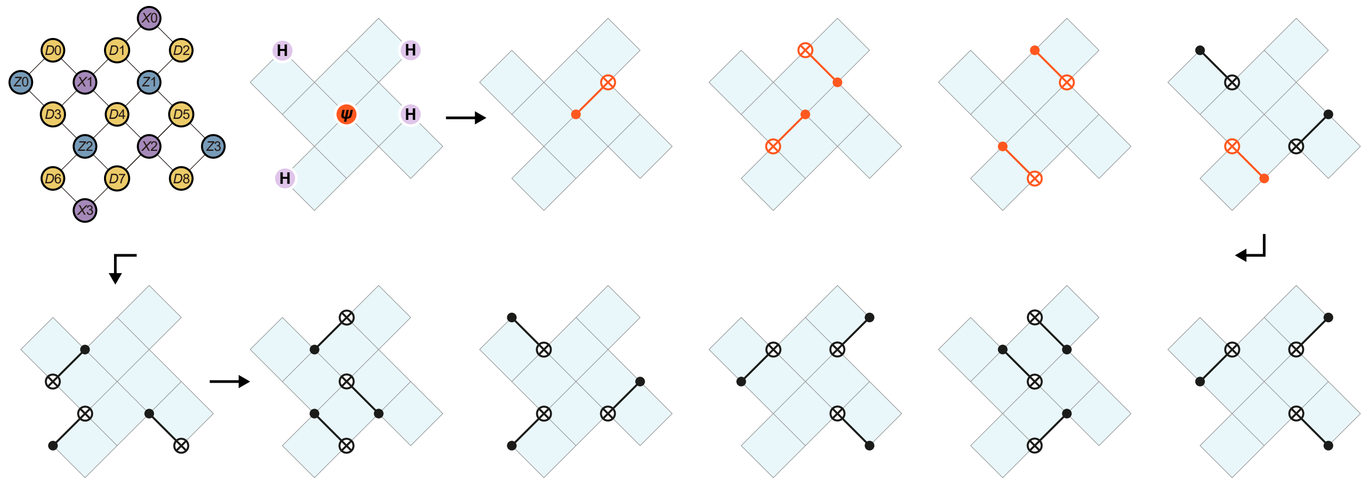
Round, M	1	2	3	4
Unit error probability, p (%)	13.6	9.4	7.2	5.7



Supplementary Fig. 5. **Relative bias and sampling overhead of the ZNE.** For the implementation of ZNE, we use $K = 1$. Note that δ/δ_0 is presented in a linear scale, whereas η is shown in a logarithmic scale. Inset: biases without ZNE.

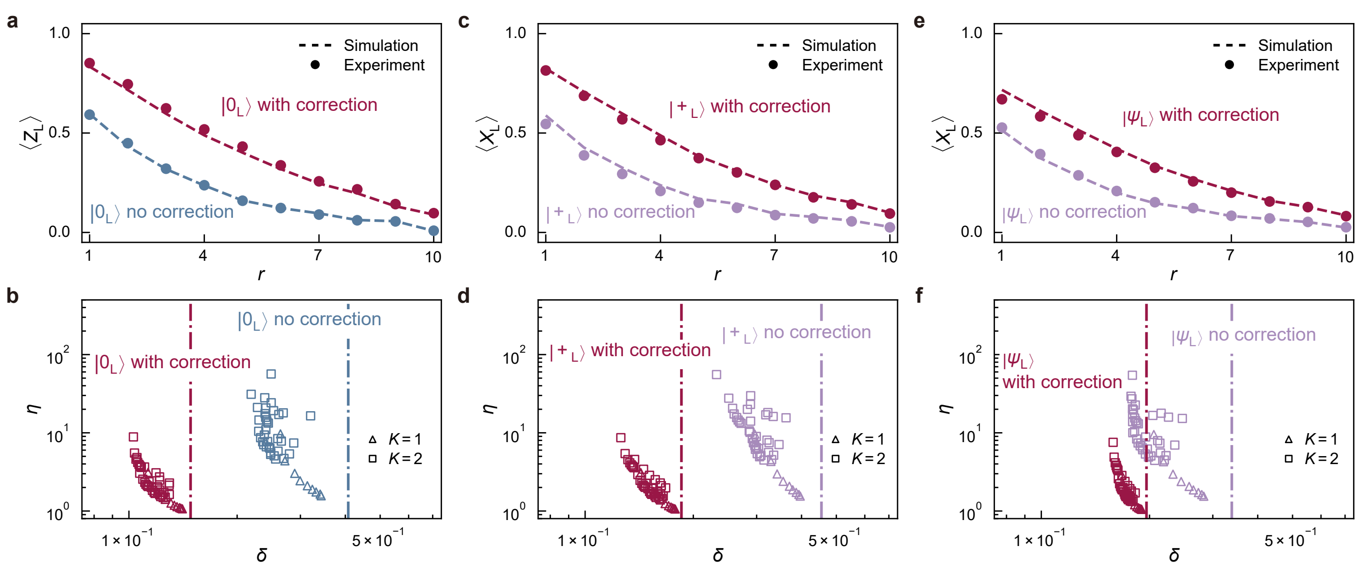
Preparation of an arbitrary logical state in the distance-3 surface code. In this work, an arbitrary logical state in the distance-3 surface code is prepared using the circuit illustrated in Supplementary Fig. 6. For specific cases, such as the initialization of states $|0_{\mathrm{L}}\rangle$ and $|+_{\mathrm{L}}\rangle$, the circuit can be further simplified, as described in Ref. [2]. Additionally, the implementation of state initialization in a configuration where two-qubit gates can be performed between nearest-neighbor data qubits is detailed in Ref. [3].

More ZNE results in the distance-3 surface code. The expectation values of the Z_{L} and X_{L} observables are measured as a function of the noise scaling factor r for the states $|0_{\mathrm{L}}\rangle$ and $|+_{\mathrm{L}}\rangle$, respectively (Supplementary Fig. 7). These results are consistent with those obtained for $|\psi_{\mathrm{L}}\rangle$ in the main text but specifically address either bit-flip error or phase-flip error. The measured values of the X_{L} observable, with the initial state set to $|\psi_{\mathrm{L}}\rangle = \cos\frac{\pi}{6}|0_{\mathrm{L}}\rangle + \sin\frac{\pi}{6}|1_{\mathrm{L}}\rangle$,



Supplementary Fig. 6. **Circuit for the initialization of an arbitrary logical state.** The preparation of an arbitrary logical state $|\psi_{\rm L}\rangle = \alpha\,|0_{\rm L}\rangle + \beta\,|1_{\rm L}\rangle$ proceeds as follows. First, a single-qubit gate initializes data qubit $D4$ in the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, with Hadamard gates applied to four representative qubits. Next, six CNOT gates (highlighted in red) are used to create a GHZ-like state — $\alpha|000\rangle + \beta|111\rangle$ — on the vertically aligned data qubits $D1$, $D4$, and $D7$. Finally, multiple layers of CNOT gates are sequentially applied to construct the desired logical state. In particular, for the preparation of state $|0_{\rm L}\rangle$, the six CNOT gates highlighted in red are unnecessary and can be omitted.

are also shown in the figure. All δ-η scatter plots demonstrate the effectiveness of ZNE in the surface code.



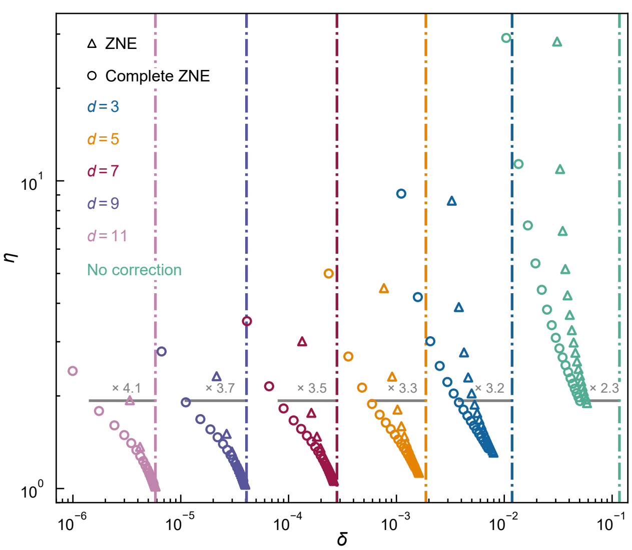
Supplementary Fig. 7. **Experimental results of ZNE for the $|0_{\rm L}\rangle$ and $|+_{\rm L}\rangle$ states in the distance-3 surface code.** **a**–**b**, Measured expectation values of the $Z_{\rm L}$ observable for the logical state $|0_{\rm L}\rangle$ and corresponding scatter plots of the bias δ and sampling overhead η. **c**–**f**, Similar to panels **a** and **b**, but with the observable being $X_{\rm L}$ and the initial states being $|+_{\rm L}\rangle$ (panels **c** and **d**) and $|\psi_{\rm L}\rangle = \cos\frac{\pi}{6}|0_{\rm L}\rangle + \sin\frac{\pi}{6}|1_{\rm L}\rangle$ (panels **e** and **f**).

Numerical simulations. We conduct numerical simulations using two popular open-source frameworks: Stim [4] and Qiskit [5]. For simulations of the repetition code, where the initial preparation of the logical state ($|0_{\rm L}\rangle = |0\cdots0\rangle$) is nearly perfect, errors introduced by the gate and measurement imperfections are then modeled using a depolarizing channel and a bit-flip channel, respectively. The error rates for these models are experimentally calibrated (see Fig. 5 in the main text and Supplementary Table 1). In contrast, for simulations of the surface code, where initial states are imperfectly generated via multiple layers of single-qubit and two-qubit gates, the introduced imperfection is simplified by a single-qubit depolarizing channel uniformly applied to each qubit. The depolarizing rate is calibrated by matching the expectation value of an observable with error injection disabled in the circuit. In this work, $|0_\mathrm{L}\rangle$ is used as the initial state, and Z_L as the observable, resulting in a calibrated depolarizing rate of 0.075. As shown in Fig. 4 of the main text, this noise model closely aligns with the experimental data.

Supplementary Note 5. PROTOCOL

A. Complete ZNE with error mitigation on parity-check measurement

While Fig. 3 in the main text suggests a decrease in ZNE's error suppression power with increasing code distance, this observation stems from a model simplification: we only amplifies the injected errors, given their significantly greater magnitude compared to the errors of single/two-qubit gate and state discrimination within parity check circuits. Even with this simplification, ZNE still demonstrated improved precision in measuring the expectation values. In addition, the experimental results are in close agreement with the numerical simulations (Fig. 3 in the main text). As a complementary step, we numerically perform a complete ZNE on logical qubits, amplifying all error sources—including injected errors, gate errors, and measurement errors. As shown in Supplementary Fig. 8, for a fixed sampling overhead ($\eta = 1.93$), the error suppression achieved by ZNE combined with error correction (≥ 3.2) consistently outperforms that without error correction (2.3), strongly suggesting the advantages of integrating error mitigation and error correction.



Supplementary Fig. 8. **Numerical results of error amplification on parity-check circuits.** Circles and triangles depict the bias and sampling overhead for scenarios where imperfections due to single/two-qubit gates and state discrimination within the parity-check circuits are either amplified or not. For a given sampling overhead such as $\eta = 1.93$, error suppression power of complete ZNE grows with increasing the code distance.

B. Application to large-scale logical circuits

We take the surface code as an example to illustrate that our method can be applied to logical circuits at large scale. Assume we have a logic target circuit consisting of N logical gates. Similar to physical operations (see Supplementary Note 1), the jth logical gate reads

$$\mathcal{M}_j = (1 - P_j)\mathcal{M}_j^I + P_j \mathcal{M}_j^E, \tag{22}$$

where P_j represents the logical error rate of the gate. We remark that the logical error rate depends on the physical error rate and, therefore, is a function of the noise amplification factor r.

The expected value of the observable O computed with the circuit is

$$\begin{aligned}\langle O\rangle &= \mathrm{Tr}(O\mathcal{M}_N\cdots\mathcal{M}_2\mathcal{M}_1\rho)\\ &= (1-\sum_{j=1}^{N}P_j)\langle O\rangle_{\text{ideal}}+\sum_{j=1}^{N}P_j\langle O\rangle_j+R_1,\end{aligned}\tag{23}$$

where

$$\langle O\rangle_{\text{ideal}}=\mathrm{Tr}(O\mathcal{M}_N^I\cdots\mathcal{M}_2^I\mathcal{M}_1^I\rho)\tag{24}$$

is the ideal result,

$$\langle O\rangle_j=\mathrm{Tr}(O\mathcal{M}_N^I\cdots\mathcal{M}_j^E\cdots\mathcal{M}_2^I\mathcal{M}_1^I\rho)\tag{25}$$

is the result when the jth logical gate has errors, and the remainder term has the upper bound

$$\begin{aligned}R_1 &\leq \|O\|\left[\prod_{j=1}^{N}(1+2P_j)-(1+2\sum_{j=1}^{N}P_j)\right]\\ &\leq \|O\|\left[e^{2P_{tot}}-(1+2P_{tot})\right],\end{aligned}\tag{26}$$

where $P_{tot}=\sum_{j=1}^{N}P_j$ is the total error rate.

Now, we approximate each logical error rate with a polynomial, i.e.

$$P_j(r)=\sum_{k=\lceil d/2\rceil}^{\lceil d/2\rceil+K-1}a_{j,k}r^k+\Delta_j(r),\tag{27}$$

and $\Delta_j(r)$ denotes the error in the approximation. With this approximation, we can re-express the expected value as

$$\begin{aligned}\langle O\rangle(r)=\langle O\rangle_{\text{ideal}}&+\sum_{k=\lceil d/2\rceil}^{\lceil d/2\rceil+K-1}a_kr^k\\ &+R_1(r)+R_2(r),\end{aligned}\tag{28}$$

where

$$a_k=\sum_{j=1}^{N}(\langle O\rangle_j-\langle O\rangle_{\text{ideal}})a_{j,k}\tag{29}$$

and

$$R_2(r)=\sum_{j=1}^{N}(\langle O\rangle_j-\langle O\rangle_{\text{ideal}})\Delta_j(r).\tag{30}$$

Substituting the expression of $\langle O\rangle(r)$ into the Kth-order extrapolation formula as Eq. (3) in the Method section, we obtain the bias

$$\delta=\left|\sum_{k=0}^{K}b_k[R_1(r_k)+R_2(r_k)]\right|\leq\tilde{\delta}_1+\tilde{\delta}_2,\tag{31}$$

where

$$\tilde{\delta}_1=\|O\|\sum_{k=0}^{K}|b_k|\left[e^{2P_{tot}(r_k)}-(1+2P_{tot}(r_k))\right]\tag{32}$$

and

$$\tilde{\delta}_2=2N\|O\|\left|\sum_{k=0}^{K}b_k\Delta_j(r_k)\right|\tag{33}$$

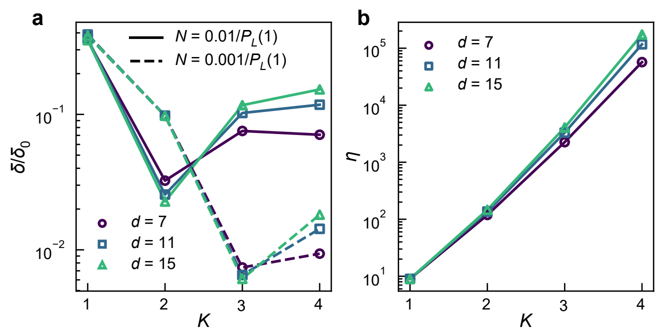
are upper bounds of contributions by R_1 and R_2 terms, respectively. The first term $\tilde{\delta}_1$ is the error due to the second-order contribution of logical error rates, i.e.

$$\tilde{\delta}_1 \simeq 2\|O\| \sum_{k=0}^{K} |b_k| P_{tot}(r_k)^2. \tag{34}$$

If we choose a sufficiently large code distance such that $P_{tot}(r_k)$ is much smaller than one, the first term is much smaller than the raw bias before error mitigation, which has the upper bound

$$\tilde{\delta}_0 = \|O\| \left[e^{2P_{tot}(1)} - 1\right] \simeq 2\|O\| P_{tot}(1). \tag{35}$$

The second term $\tilde{\delta}_2$ is the error due to approximating logical error rates $P_j(r)$ with polynomials. In the limit that the physical error rate p approaches zero, the logical error rate behaves as $P_j(r) \propto p^{\lceil d/2 \rceil}$ [6, 7], suggesting that the $K=1$ extrapolation is sufficient in the low physical error rate regime. When p is finite, the logical error rate deviates from $P_j(r) \propto p^{\lceil d/2 \rceil}$, and we have to adapt the extrapolation function accordingly. Next, we use numerical calculations to estimate the bias in the polynomial extrapolation.



Supplementary Fig. 9. **Numerical results of ZNE on a circuit of N surface-code idle gates.** We take the physical error rate $p = 10^{-3}$, which is amplified to $r_k p$ in ZNE; we choose the amplification factors $r_k = k^{1/\lceil d/2 \rceil}$, where $k = 1, 2, \ldots, K+1$. The bias before ZNE is $\delta_0 = |\langle O \rangle(1) - \langle O \rangle(0)|$, where $\langle O \rangle(0)$ is the ideal value.

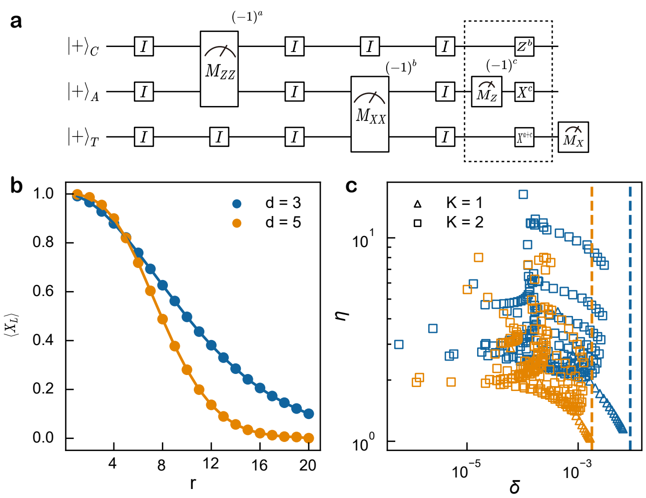
To estimate the bias and cost numerically, we consider a quantum memory circuit consisting of applying N idle gates on a logical qubit and take the logical error rate per gate reported in Ref. [8]. We choose the observable as Y, such that the expectation value is affected by both logical X and Z errors. Let $P_L(r)$ be the sum of logical X and Z error rates. The observable expectation value is taken as $\langle O \rangle(r) = [1 - 2P_L(r)]^N$. Then, we apply ZNE to $\langle O \rangle(r)$, and the results are shown in Supplementary Fig. 9.

Although the above numerical results are obtained from quantum memory circuits, we conjecture that general computing circuits yield similar results. Memory circuits are composed of repeatedly applying parity-check measurements on a fixed lattice. Practical protocols of quantum computing in surface codes include, for instance, braiding transformation [9] and lattice surgery [10], in which circuits are also composed of repeatedly applying parity-check measurements, however, on a lattice that deforms between certain parity-check cycles. Because the dominant operations are the same in memory circuits and such computing circuits, they should yield similar behavior of logical error rates (as functions of physical error rates and the code distance). In addition to braiding transformation and lattice surgery, universal quantum computing in surface codes also requires magic state injection and distillation [9, 11, 12]. Magic state errors decrease with the level of distillation. As long as the distillation level is adequately high, the post-distillation magic state errors are dominated by logical errors in lattice surgery operations, and the contribution of raw magic state errors (errors in injected magic states before distillation) is negligible; otherwise, we may need to modify the extrapolation function to include the impact of raw magic state errors: suppose the distilled magic state error rate is $\propto p^L$, where L is an integer depending on the distillation level, we can add the term p^L to the extrapolation function.

So far, we have only considered polynomial extrapolation functions. The use of other extrapolation functions could improve the performance. In ZNE on NISQ circuit, it has been shown experimentally that the exponential extrapolation outperforms polynomial extrapolation [13, 14]. For example, the logical error rate function reported in Ref. [8] is a candidate worth exploring. Given such candidate functions, we can benchmark and verify them through Clifford circuits [15, 16]: we can simulate the Clifford circuits on a classical computer to obtain the ideal result $\langle O \rangle_{\mathrm{ideal}}$, such that we can evaluate the bias as well as the cost. In this way, we can compare the candidate the functions and identify the suitable ones.

C. Application to logical computation circuits

We simulate a CNOT gate implemented via lattice surgery [10, 17] whthin three surface code blocks to demonstrate the effectiveness of our method in logical quantum circuits. To better reflect the noise characteristics of actual quantum devices, we set the original noise levels (i.e., at noise scaling factor $r = 1$) as depolarizing error rates of 0.0001, 0.0001, 0.0015, and 0.0030 for initialization, idle gates, two-qubit gates, and measurements, respectively. Although these original noise levels are not compatible with existing devices, they are expected to be achievable in the near future. These error rates are then uniformly scaled by varying r. Starting with both logical qubits initialized in the $|+_{\mathrm{L}}\rangle$ state, we measure the $\langle X_{\mathrm{L}}\rangle$ eigenvalue of the target logical qubit after a CNOT gate. The results are shown in Supplementary Fig. 10**b**. We collected data at 19 different scaled noise levels and performed first-order extrapolation ($K = 1$) using each scaled result paired with the original one. Additionally, we performed second-order extrapolation ($K = 2$) by combining the original result with all possible result pairs of scaled noise levels, as shown in Supplementary Fig. 10**c**. The results indicate that higher-order extrapolation can achieve lower bias, though this comes with higher resource costs, which is consistent with the analysis in Supplementary Note 5 B. In some cases, the second-order ZNE performs worse than the first-order due to its sensitivity to statistical fluctuations. This issue can be mitigated by increasing the number of measurements or by optimizing the choice of noise scaling points.



Supplementary Fig. 10. **Numerical results of ZNE on a logical CNOT gate circuit under circuit-level noise.** **a**, Measurement-based CNOT circuit diagram. The control qubit (C), ancilla qubit (A), and target qubit (T) are initially prepared in the $|+\rangle$ state. They then undergo a logical CNOT operation, which consists of joint measurements (M_{ZZ}, M_{XX}) and identity gates (I), followed by an eigenvalue measurement of the X_{L} operator on the target qubit. For simplicity, some post-processing operations (indicated by the dashed box) based on joint measurement outcomes are omitted in our simulation since these operations do not affect the measurement results of the observable X_{L} [17]. **b**, Measured expectation values of the X_{L} observable with respect to the noise scaling factor r in circuits. **c**, Scatter plot showing the bias δ and sampling overhead η for all possible choices of $r_1, \ldots, r_K$ in ZNE.

* These authors contributed equally.
† pfzhang@zju.edu.cn
‡ yli@gscaep.ac.cn

[1] S. Bravyi, A. W. Cross, J. M. Gambetta, D. Maslov, P. Rall, and T. J. Yoder, High-threshold and low-overhead fault-tolerant quantum memory, Nature **627**, 778 (2024).
[2] Y. Zhao, Y. Ye, H.-L. Huang, Y. Zhang, D. Wu, H. Guan, Q. Zhu, Z. Wei, T. He, S. Cao, F. Chen, T.-H. Chung, H. Deng, D. Fan, M. Gong, C. Guo, S. Guo, L. Han, N. Li, S. Li, Y. Li, F. Liang, J. Lin, H. Qian, H. Rong, H. Su, L. Sun, S. Wang,

Y. Wu, Y. Xu, C. Ying, J. Yu, C. Zha, K. Zhang, Y.-H. Huo, C.-Y. Lu, C.-Z. Peng, X. Zhu, and J.-W. Pan, Realization of an error-correcting surface code with superconducting qubits, Phys. Rev. Lett. **129**, 030501 (2022).

[3] K. J. Satzinger, Y.-J. Liu, A. Smith, C. Knapp, M. Newman, C. Jones, Z. Chen, C. Quintana, X. Mi, A. Dunsworth, C. Gidney, I. Aleiner, F. Arute, K. Arya, J. Atalaya, R. Babbush, J. C. Bardin, R. Barends, J. Basso, A. Bengtsson, A. Bilmes, M. Broughton, B. B. Buckley, D. A. Buell, B. Burkett, N. Bushnell, B. Chiaro, R. Collins, W. Courtney, S. Demura, A. R. Derk, D. Eppens, C. Erickson, L. Faoro, E. Farhi, A. G. Fowler, B. Foxen, M. Giustina, A. Greene, J. A. Gross, M. P. Harrigan, S. D. Harrington, J. Hilton, S. Hong, T. Huang, W. J. Huggins, L. B. Ioffe, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, T. Khattar, S. Kim, P. V. Klimov, A. N. Korotkov, F. Kostritsa, D. Landhuis, P. Laptev, A. Locharla, E. Lucero, O. Martin, J. R. McClean, M. McEwen, K. C. Miao, M. Mohseni, S. Montazeri, W. Mruczkiewicz, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Y. Niu, T. E. O'Brien, A. Opremcak, B. Pató, A. Petukhov, N. C. Rubin, D. Sank, V. Shvarts, D. Strain, M. Szalay, B. Villalonga, T. C. White, Z. Yao, P. Yeh, J. Yoo, A. Zalcman, H. Neven, S. Boixo, A. Megrant, Y. Chen, J. Kelly, V. Smelyanskiy, A. Kitaev, M. Knap, F. Pollmann, and P. Roushan, Realizing topologically ordered states on a quantum processor, Science **374**, 1237 (2021).

[4] C. Gidney, Stim: a fast stabilizer circuit simulator, Quantum **5**, 497 (2021).

[5] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta, Quantum computing with Qiskit (2024), arXiv:2405.08810 [quant-ph].

[6] A. G. Fowler, Analytic asymptotic performance of topological codes, Phys. Rev. A **87**, 040301 (2013).

[7] F. H. E. Watson and S. D. Barrett, Logical error rate scaling of the toric code, New J. Phys. **16**, 093045 (2014).

[8] S. Bravyi and A. Vargo, Simulation of rare events in quantum error correction, Phys. Rev. A **88**, 062308 (2013).

[9] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Surface codes: Towards practical large-scale quantum computation, Phys. Rev. A **86**, 032324 (2012).

[10] D. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, Surface code quantum computing by lattice surgery, New J. Phys. **14**, 123011 (2012).

[11] S. Bravyi and A. Kitaev, Universal quantum computation with ideal clifford gates and noisy ancillas, Phys. Rev. A **71**, 022316 (2005).

[12] R. Raussendorf, J. Harrington, and K. Goyal, Topological fault-tolerance in cluster state quantum computation, New J. Phys. **9**, 199 (2007).

[13] S. Endo, S. C. Benjamin, and Y. Li, Practical quantum error mitigation for near-future applications, Phys. Rev. X **8**, 031027 (2018).

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

[15] P. Czarnik, A. Arrasmith, P. J. Coles, and L. Cincio, Error mitigation with Clifford quantum-circuit data, Quantum **5**, 592 (2021).

[16] A. Strikis, D. Qin, Y. Chen, S. C. Benjamin, and Y. Li, Learning-based quantum error mitigation, PRX Quantum **2**, 040330 (2021).

[17] C. Vuillot, L. Lao, B. Criger, C. García Almudéver, K. Bertels, and B. M. Terhal, Code deformation and lattice surgery are gauge fixing, New Journal of Physics **21**, 033028 (2019).