Supplementary information: Physics-Inspired Machine Learning for Quantum Error Mitigation

I. THE ESTIMATION OF EFFECTIVENESS FACTORS

In the main text, we compute the mitigated expectation value at the l-th layer by utilizing the estimated effectiveness factors $\{\hat{p}_j\}_{j=1}^{l}$. These factors are incorporated within the maximum noise as detailed in the 'Methods' section of the main text, serving as thresholds to preserve the Completely Positive and Trace-Preserving (CPTP) characteristics of the effective noise channels Λ_{ll}. In this section, we derive the formula for estimating these effectiveness factors.

Consider an n-qubit noisy layer $\tilde{u}$, which is composed of D noisy gates arranged as $\tilde{u} = \tilde{G}_D \circ \ldots \circ \tilde{G}_2 \circ \tilde{G}_1$. Under the non-correlation assumption, we consider that quantum gates acting on different qubits are free from crosstalk, ensuring the independence of their respective noise channels. Each noisy gate $\tilde{G}_d$ can be decomposed into $\tilde{G}_d = \Theta_d \circ G_d$ where Θ_d is a CPTP noise channel. Through maximum noise decomposition, we can derive the expression of Θ_d as $\Theta_d = (1-p^d)\mathcal{I} + p^d\Lambda^d$. To calculate the effectiveness factor p, we shift the gate noise Θ_d backward:

$$\begin{aligned}
\tilde{G}_{d+1} \circ \Theta_d &= \Theta_{d+1} \circ G_{d+1} \circ \Theta_d \\
&= \Theta_{d+1} \circ (G_{d+1}\Theta_d G_{d+1}^{\dagger}) \circ G_d \\
&= \Theta_{d+1} \circ (G_{d+1}[(1-p^d)\mathcal{I} + p^d\Lambda^d]G_{d+1}^{\dagger}) \circ G_d \\
&\equiv \Theta_{d+1} \circ \Theta_d^{G_{d+1}} \circ G_d,
\end{aligned} \tag{1}$$

where the factor p^d of the noise channel $\Theta_d^{G_d}$ twirled by G_{d+1} remains invariant upon backward relocation. The effectiveness factor p is then computed using

$$p = 1 - \prod_{d=1}^{D}(1-p^d). \tag{2}$$

Since the layer can be decomposed into single or two-qubit gates through circuit compilation, the estimation of p^d for each gate within the layer is feasible via quantum process tomography [1] techniques. Specifically, for the Pauli gate noise model, p^d can be efficiently estimated through randomized benchmarking [2], as we utilize in our simulations involving the Pauli noise model. Experimental results in Fig. S1 for Quantum Approximate Optimization Algorithm (QAOA)-type circuits under stochastic Pauli noise model indicate that our model shows resilience in estimating prior value p, displaying a moderate level of tolerance.

II. QUANTUM ERROR MITIGATION METHODS IN BASELINES

In this section, we will provide the details for typical quantum error mitigation (QEM) strategies that serve as baselines in the 'Results' section of main text, along with the precise parameter settings employed in their implementation.

A. Standard quantum error mitigation methods

The standard QEM methods we employ include the zero-noise extrapolation (ZNE), probabilistic error cancellation (PEC) and Clifford Data Regression (CDR). To ensure fair benchmarking across different QEM methods, we rigorously control the total measurement shots allocated to each technique. For ZNE, CDR, and the noisy case (Noisy), we fix the number of shots per circuit instance to $N_S = 8192$. For PEC, due to its circuit-sampling nature, it deviates from the fixed-per-circuit-instance shot model used by the aforementioned methods. However, we adjust for this by ensuring PEC's total measurement shots amount to $N_T = 8192 * 2$, matching ZNE's total and being twice that of the noisy case. It is worth noting that while exponentially increasing measurement shots can enhance the precision of standard QEM methods, the primary objective with machine learning for quantum error mitigation (ML-QEM) is to mitigate the sampling overhead. The shot configuration in our experiments aligns with typical algorithmic scenarios, ensuring a balance between precision and resource efficiency.

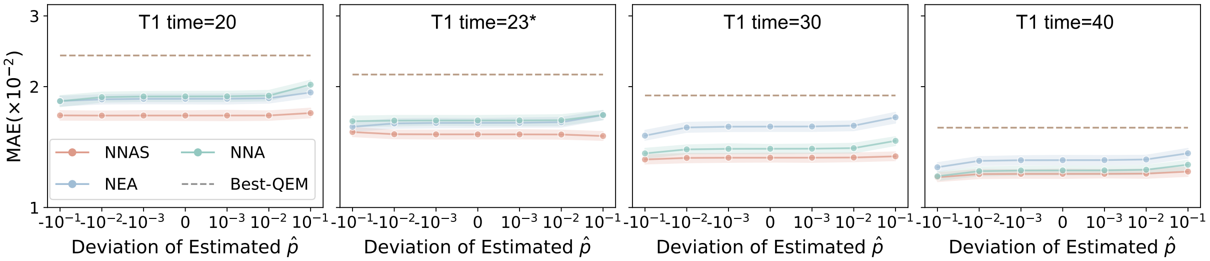

FIG. S1. **The performance of our model and the ablated models using estiamted value $\hat{p}$ with varying degrees of deviation from the exact value of p, across varying noise levels.** Longer T1 time denotes lower noise level. We focus on T1 = 23.235 μs which represents the state-of-the-art quantum coherence times. Results indicates that the impact of the deviation pf $\hat{p}$ from the exact value on the precision of our model's performance is negligible, indicating a robustness of the estimation errors. All the data in the figure presents average results with 95% confidence interval error bars.

1. *Zero-Noise Extrapolation*

ZNE is a canonical QEM technique that leverages the principle of extrapolating noisy expectation values to estimate the noiseless value. This method is particularly adept at handling errors that arise from the quantum hardware itself. ZNE operates by scaling the noise levels in a controlled manner and then extrapolating the noisy results to the zero-noise limit. We hypothesize the ability to scale the noise level to integer multiples of the base noise level ϵ, with the multiples $1 = x_0 < x_1 < ... < x_r$. We execute the circuits under these amplified noise levels $\{x_i \epsilon\}_{i=0}^{r}$ and collect the corresponding noisy expectation values $\{\langle \tilde{O} \rangle_i\}_i$. Next, we model the expectation values as a function of ϵ, and then extrapolate this function to estimate the noise-free limit. The mitigated expectation value is calculated using Richardson extrapolation, which employs a linear equation to extrapolate the noise-free limit. The calculation is

$$\langle O \rangle_{\text{ZNE}} = \sum_{i=0}^{r} \gamma_i \langle \tilde{O} \rangle_i, \tag{3}$$

where the fitting coefficients $\{\gamma_i\}_i$ meet $\sum_{i=0}^{r} \gamma_i = 1$ and $\sum_{i=0}^{r} \gamma_i x_i^j = 0$ for $j = 1, ..., r$ and the solutions gives

$$\gamma_i = \prod_{i \neq j} \frac{x_j}{x_j - x_i}. \tag{4}$$

We implement two-point noise amplification and apply Gate Unfolding [3] as our method for noise scaling. Specifically, within the noise-amplified setup, each multi-qubit gate in the circuit is applied $2m+1$ times, with $2m+1$ representing the amplification factor. We have chosen $m = 1$, corresponding to a threefold amplification of the noise level. In our setting, the implementation of ZNE involves two circuit instances: one for the circuit under the original noise level and another for the circuit with noise amplified threefold.

2. *Probabilistic Error Cancellation*

PEC is one of the earliest QEM algorithms proposed. Its fundamental concept involves calibrating the noisy process channel expression to implement the inverse of the noise channel, which is then appended to the noisy circuit. We utilize the PEC method with sparse Pauli-Lindblad models (SPL) [4, 5]. Given that single-qubit Pauli gates, which are intended for noise cancellation, are themselves susceptible to noise contamination, their effectiveness in mitigating noise for single-qubit gates is limited. In contrast, the noise impact of two-qubit gates is significantly greater than that of single-qubit gates. Consequently, the additional noise introduced by single-qubit gates is negligible compared to the noise reduction achieved by error mitigation for two-qubit gates. Based on this analysis, our focus is primarily on mitigating the impact of noisy two-qubit gates. For a two-qubit noisy gate, the SPL model is expressed as:

$$\mathcal{E}(\cdot) = \prod_{k} \left(\omega_k \mathcal{I} + (1 - \omega_k) \mathcal{P}_k \right), \tag{5}$$

where $\omega_k = \frac{1+e^{-2\lambda_k}}{2}$ and $\{\lambda_k\}_k$ are the noise coefficients of the SPL model, k ranges from 0 to $4^2 - 1$. Then we need to derive the inverse expression of $\mathcal{E}$ to implement PEC. This derivation is notably efficient, as it simply involves negating the noise coefficients $\{\lambda_k\}_k$. This straightforward approach offers a significant advantage in that it circumvents the need for complex inversion of the full noise channel $\mathcal{E}$, which often leads to unphysical channels. These noise coefficients are related to the diagonal entries of the Pauli Transfer Matrix (PTM) of the SPL noise channel $\mathcal{E}$, which can be linked to the channel's impact on the Pauli operator P_a with fidelity f_a:

$$f_a = \frac{1}{2^n}\mathrm{Tr}[P_a^\dagger \mathcal{E}(P_a)] = \frac{1}{2^n}\mathrm{Tr}[P_a^\dagger P_a \prod_{\{a,k\}=0}(2\omega_k - 1)] = e^{-2\sum_{\{a,k\}=0}\lambda_k}, \tag{6}$$

where $\{a,k\} = 0$ denotes that the Pauli operators P_a and P_k anti-commute. We calculate the noise coefficients $\{\lambda_k\}_k$ via

$$-\log(f)/2 = M\lambda, \tag{7}$$

where M is a matrix characterizing the commutativity for Pauli operators, with the entry $M_{a,b} = 0$ representing that Paulis P_a and P_b commute and $M_{a,b} = 1$ otherwise. The Pauli fidelity, denoted as f_a, can be determined by fitting the decay function $a_0\hat{f}_a^m$. This approach effectively mitigates the impact of state preparation and measurement (SPAM) errors. Here, m denotes the number of repetitions of the noisy multi-qubit gates. In our experiment setting, the multi-qubit gate consists solely of the CNOT gate, which is self-conjugate. We select an even set of m values, specifically $m = \{2, 4, 6, 8, 10\}$, to fit the decay function.

After determining the noise coefficients $\{\lambda_k\}_k$, we calculate the set $\{\omega_k\}_k$ and derive the full inverse channel using:

$$\mathcal{E}^{-1}(\cdot) = \gamma\prod_k(\omega_k\mathcal{I} - (1-\omega_k)\mathcal{P}_k), \tag{8}$$

where γ is the sampling overhead:

$$\gamma = \prod_k(2\omega_k - 1) = e^{\left(\sum_k 2\lambda_k\right)}. \tag{9}$$

We then append the inverse channel $\mathcal{E}^{-1}$ to the noisy gate, employing the quasi-probability decomposition [4]. For each k, we sample the identity $\mathcal{I}$ with probability ω_k and the Pauli channel P_k with probability $1-\omega_k$, and subsequently append it to the noisy gate. We keep track of the sign (-1) each time a Pauli channel is sampled. This procedure is repeated for each two-qubit gate in the circuit, and all the signs accumulate. Finally, we adjust the outcomes by multiplying $(-1)^m\gamma = (-1)^m\prod_{j=1}^{G}\gamma_i$ across G two-qubit gates, where m is the number of chosen Pauli channels and γ_i is the sampling overhead for each gate, calculated using Eq. 9. In our setting, we sample 200 random circuit instances for a complete mitigation process, following the protocol established in Ref. [4]. The measurement shots used for calibration of noise parameters ω_k are 10^5 per circuit instance and are separate from those allocated for PEC mitigation.

3. *Clifford Data Regression*

CDR is a typical learning-based QEM method, which can be regarded as the simplest form of linear model-based machine learning quantum error mitigation (ML-QEM) method. CDR differs from methods like PEC, which depend on exact knowledge of the noise channel, and ZNE, which requires precise control over the noise level. Instead, CDR estimates the expectation values of target circuits by establishing a linear relationship between noisy and noiseless expectation values. This relationship is learned using training datasets derived from near-Clifford circuits that are relevant to the target circuit. These near-Clifford circuits, which are classically simulatable, are constructed by substituting some non-Clifford gates in the target circuit with Clifford gates.

Our training dataset comprises near-Clifford circuits with a replacement rate of $r_p = 0.5$. The non-Clifford gates we utilized include the X-rotation gate $R_x(\theta)$, the Y-rotation gate R_y, and the Z-rotation gate R_z. We replace a proportion of r_p of these gates with single-qubit Clifford gates according to the following replacement rule,

$$\begin{aligned} R_x(\theta) &\Longrightarrow R_x(\frac{k\pi}{2}), \\ R_y(\theta) &\Longrightarrow R_y(\frac{k\pi}{2}), \\ R_z(\theta) &\Longrightarrow R_z(\frac{k\pi}{2}), \end{aligned} \tag{10}$$

TABLE I. The setting of grid search.

number of estimators		maximum depth			minimum samples split		minimum samples leaf		maximum features	
min	max	option	min	max	min	max	min	max	number	option
100	200	None	10	20	2	5	1	2	1	sqrt

where k is randomly sampled with the weight computed by

$$w(k) = e^{-4d^2}, d = \|R_{x,y,z}(\theta) - R_{x,y,z}(\frac{k\pi}{2})\|. \tag{11}$$

For each target circuit, we generate and measure 10 corresponding near-Clifford circuits for training, each containing a maximum of 20 non-Clifford gates. These measurements are conducted using the identical observable as that of the target circuit.

B. Data-driven machine learning quantum error mitigation methods

We choose the typical data-driven machine learning model random forest to mitigate noisy circuits, and we refer to this approach as RF. RF is an ensemble machine learning method that integrates the prediction of multiple decision trees. Comparing with a single decision tree, the RF method exhibits a preferable capacity against over-fitting, but it also entails greater consumption of computing resources. We utilize the random forest regression algorithm following [6] for QEM, with the RandomForestRegressor as a RF implement and the GridSearchCV as an exhaustive parameter search. The setting of parameter searching grid are as table I.

C. Our models including ablation models

In the main text, we use two ablation models: the neural extractor ablated (NEA) and the quantum-inspired neural network ablated (NNA). The two models, NEA and NNA, are both founded on the principle of isolating the noise envelope R_l, aiming to the estimated noise impact factor $\hat{r}_l$.

1. *Attention-core ablated model*

The architecture of the NEA model is rooted in the neural noise accumulator, which is realized by a Recurrent Neural Network (RNN). In our model, we integrate circuit features and concatenate them to form the input data matrix $X \in \mathbb{R}^{L \times M}$, where L denotes the sequence length, and M represents the number of variables. The RNN within our model consists of a hidden state H that captures information from prior inputs, and an output Y, which is a sequence of the predicted results. At each time step l, the hidden state H_l updates recursively according to the following equation:

$$H_l = f(H_{l-1}, X_l) = \tanh(W_1 H_{l-1} + b_1 + W_2 X_l + b_2), \tag{12}$$

where W_1, b_1, W_2, b_2 are learnable parameters. Subsequently, a readout layer, consisting of a linear transformation, is employed to extract the noise component Y_l:

$$Y_l = g(H_l) = W_3 H_l + b_3, \tag{13}$$

with W_3, b_3 being additional learnable parameters. Ultimately, the desired output Y_l^* is computed as:

$$Y_l^* = \frac{\langle \tilde{O} \rangle_l}{\prod_{j=1}^{l}(1 - \hat{p}_j) + Y_l}. \tag{14}$$

Here, Y_l^* represents the predicted results.

2. *Quantum-inspired neural network ablated*

We then explore the ablation of the entire physics-inspired neural architecture by employing a standard artificial intelligence approach: the Multilayer Perceptron (MLP). The MLP utilized in the NNA consists of multiple hidden layers, each defined by the following linear transformation,

$$X^{(i)} = W^{(i)} X^{(i-1)} + b^{(i)}, \tag{15}$$

where $X^{(i)} \in \mathbb{R}^{L\times M}$ represents the output of the i-th layer, $W^{(i)}$ is the learnable weight matrix for the i-th layer, and i ranges from 1 to 5. In our experiments, the MLP comprises 5 layers, with each hidden layer having a size of 32 units. The input to the first layer, X^0, consists of embedded features $X \in \mathbb{R}^{L\times D}$ derived from the original input data, which includes device and circuit specifications along with noisy results. The final output of the network, $Y \in \mathbb{R}^{L\times 1}$, is produced by the last layer. Here, L denotes the length of the sequence, and D represents the dimensionality of the input features.

III. DATASET GENERATION

The dataset utilized in the main text originates from noisy simulations conducted using Qiskit [7]. We will provide a detailed exposition of the dataset's generation process and its key characteristics.

A. Gate noise model used for simulation

In our numerical simulations to generate datasets, we adopt a noise model derived from the principles of decoherence in superconducting systems, a prevalent architecture in quantum computing. Specifically, we identify amplitude damping (AD) and phase damping (PD) as the primary sources of decoherence, which are commonly characterized by the system's relaxation time T_1 and dephasing time T_2 [8, 9]. For a single qubit, the AD channel has the form

$$\mathcal{E}_{\text{AD}}(\rho) = E_1^{\text{AD}} \rho E_1^{\text{AD}^\dagger} + E_2^{\text{AD}} \rho E_2^{\text{AD}^\dagger}, \tag{16}$$

where E_1^{AD} and E_2^{AD} are defined as

$$E_1^{\text{AD}} = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-p_{\text{AD}}} \end{pmatrix} \text{ and } E_2^{\text{AD}} = \begin{pmatrix} 0 & \sqrt{p_{\text{AD}}} \\ 0 & 0 \end{pmatrix} \tag{17}$$

which are the Kraus operators of the AD noise channel $\mathcal{E}_{\text{AD}}$ with error rate p_{AD}. As for PD noise channel, the Kraus operators E_1^{PD} and E_2^{PD} are

$$E_1^{\text{PD}} = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-p_{\text{PD}}} \end{pmatrix} \text{ and } E_2^{\text{PD}} = \begin{pmatrix} 0 & \sqrt{p_{\text{PD}}} \\ 0 & 0 \end{pmatrix} \tag{18}$$

To effectively simulate these sources of noise, one can employ the randomized compiling (RC) [10–12] technique. The RC technique is utilized to map the complex effects of AD and RD directly onto a more computationally tractable Pauli noise channel. In our case, we adopt the Pauli noise model as a direct outcome of applying the RC technique to the AD and PD noise channels, which can be

$$\mathcal{E}^{\text{P}}(\rho) = (1 - p_X - p_Y - p_Z)\rho + p_X X\rho X + p_Y Y\rho Y + p_Z Z\rho Z,$$

where the parameters of the Pauli noise model are determined by the decoherence rates T_1 and T_2:

$$p_X = p_Y = \frac{1-e^{-t/T_1}}{4} \text{ and } p_Z = \frac{1-e^{-t/T_2}}{2} - \frac{1-e^{-t/T_1}}{4}.$$

We employ the average $T_1 = 23.2357\mu$s and $T_2 = 15.6\mu$s from Ref [13] as a baseline for scaling the noise proportionally, while maintaining the ratio between them. Under the assumption of uncorrelated errors, the Pauli error probabilities for two-qubit Pauli noise channel can be quantified as follows [8]:

$$\begin{aligned}
&p_{IX} = p_{XI} = p_{IY} = p_{YI} = p_X(1 - p_X - p_Y - p_Z), \\
&p_{XX} = p_{XY} = p_{YY} = p_{YX} = p_X p_Y, \\
&p_{XZ} = p_{ZX} = p_{YZ} = p_{ZY} = p_X p_Z, \\
&p_{IZ} = p_{ZI} = p_Z(1 - p_X - p_Y - p_Z), \\
&p_{ZZ} = p_Z p_Z.
\end{aligned}$$

For the gate execution time t, we set the single-qubit gate time to 18 ns and the two-qubit gate time to 48 ns, corresponding to their respective Pauli error rates. Moreover, we have incorporated a fluctuation of 2×10^{-5} to emulate the intrinsic numerical variations observed in actual quantum computing environments. The Pauli error rate, which is the aggregate of all individual Pauli error probabilities, across varying noise levels is detailed in Table III A.

TABLE II. Pauli error rates across different noise levels.

noise level		single-qubit Pauli error rate[a]	two-qubit Pauli error rate[b]
T1 time	T2 time		
20.00	13.43	0.2384%	0.4761%
23.24	15.60	0.2052%	0.4100%
30.00	20.14	0.1590%	0.3177%
40.00	26.86	0.1193%	0.2384%

[a] is calculated by 1-$p_X - p_Y - p_Z$.
[b] is calculated by 1-$(1 - p_X - p_Y - p_Z)^2$ under the uncorrelated noise assumption.

B. Metric for dataset acquisition difficulty

In the main text, we denote the *hard regime* for training dataset which represents the data points obtained from deeper circuits. In the quantum sequential task, the *hard regime* can be characterized by the maximum sequence length L for each circuit parameter. The maximum length L is determined by the selection rates listed in Table IV for QAOA-like circuit tasks and Table V for quantum metrology, which are influenced by the partial training rate p_r. Consequently, the number of data points incorporated into the training set can be calculated with L and the training data size M which is the number of chosen set of circuit parameters. For instance, the number of data points for QAOA-type circuit tasks within a 100 training size training set and $p_r = 0.25$ is $100 \cdot 10 \cdot 0.75 + 100 \cdot 13 \cdot 0.25 \cdot 0.5 + 100 \cdot 17 \cdot 0.25 \cdot 0.3 + 100 \cdot 20 \cdot 0.25 \cdot 0.2 = 1140$.

C. Experimental settings for quantum algorithms based on the Quantum Approximate Optimization Algorithm-type circuits

The QAOA-type circuits are commonly utilized in near-term quantum algorithms such as quantum Trotterized simulation and variational quantum algorithms (VQAs). In the main text, we employ the Hamiltonian which determines the temporal evolution of the 1D transverse-field Ising spin chain. The Hamiltonian is,

$$H_{\text{Ising}} = -J \sum_{j=0}^{n-2} Z_j Z_{j+1} + h \sum_{j=0}^{n-1} X_j \equiv H^Z_{\text{Ising}} + H^X_{\text{Ising}}, \tag{19}$$

here we define $H^Z_{\text{Ising}} = -J \sum_j Z_j Z_{j+1}$ and $H^X_{\text{Ising}} = \sum_j X_j$. J and h are the parameters which denote the coupling strength between neighboring spins and the transverse magnetic field, respectively. For a certain pair of circuit parameters (J, h), we measure the Trotter circuit $U_l^{J,h}$ at each step l ($l = 1, 2, ..., L$) with single Z observable Z_n on the last n-th qubit. To implement the Trotter circuit $U_l^{J,h}$, we employ the first-order Trotterized decomposition to approximate the time-evolution operator,

$$e^{-iH_{\text{Ising}}t} = e^{-i(H^Z_{\text{Ising}} + H^X_{\text{Ising}})t} \approx e^{-iH^Z_{\text{Ising}}t} e^{-iH^X_{\text{Ising}}t}. \tag{20}$$

TABLE IV. The selection rate for maximum Trotter step L.

10	11-13	14-17	18-20
1-p_r[a]	$p_r \cdot 0.5$	$p_r \cdot 0.3$	$p_r \cdot 0.2$

[a] denotes the partial training rate.

TABLE V. The selection rate for maximum number of qubits $n_{\max}$.

5	6-8	9-10
1-p_r	$p_r \cdot 0.7$	$p_r \cdot 0.3$

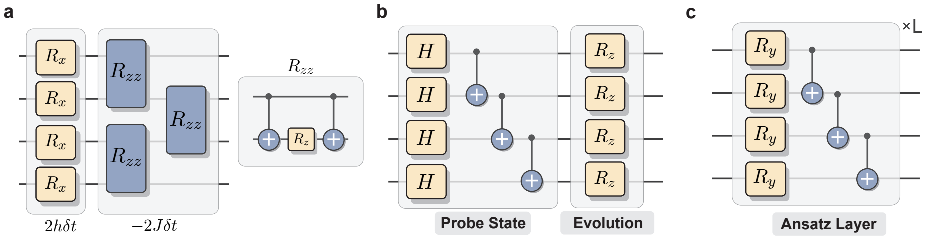

FIG. S2. **The quantum circuit for structured circuits used in numerical simulations. a** The Trotterized circuit layer with parameters (J, h), which comprises a column of X-rotation gates at $2h\delta t$ and a chain of R_{ZZ} gates. The right subfigure shows the R_{ZZ} gate's decomposition into two CNOT gates and a Z-rotation gate at $-2J\delta t$. **b** The circuit used for quantum metrology, which includes the GHZ probe state preparation circuit and the evolution circuit. The evolution circuit $U(\theta)$ parameterized by θ consists of a column of Z-rotation gates at θ. **c** The hardware-efficient (HE) circuit used for training scalability experiments utilizing near-Clifford replacement. A single ansatz layer is composed of alternating trainable single-qubit layers with rotation-y gates and a fixed entangling layer, which is a fully-connected CX gate layer. For a HE circuit with L layers, the ansatz layer is repeated L times, with each layer having independent rotation angles.

For one time step δt, we have

$$\begin{aligned} e^{-iH^{Z}_{\text{Ising}}\delta t} &= \prod_j e^{-i(-J\delta t)\sigma_z^j \sigma_z^{j+1}} \\ e^{-iH^{X}_{\text{Ising}}\delta t} &= \prod_i e^{-i(h\delta t)\sigma_x^j}, \end{aligned} \tag{21}$$

where the circuit structure of a single layer is shown in Fig. S2 **a**. The circuits span Trotter steps from 1 to 20. We maintain a fixed ratio of J to h at 0.6 and sample $h\delta t$ across the interval $[0.5, 2]$.

D. Experimental settings for quantum metrology based on GHZ probe state

Quantum metrology (QM) is one of the most appealing applications of quantum techniques, which untilizes quantum resources to estimate an unknown parameter in higher precision than the classical approaches. We focus on the QM with GHZ probe state, which enables the Heisenberg scaling $\mathcal{O}(n^{-2})$ of precision with regard to the system size n, while the presicion scales as $\mathcal{O}(n^{-1})$ for seperate states which called shot-noise scaling. We expose the n-qubit GHZ probe state $|\text{GHZ}_n\rangle = \frac{1}{\sqrt{2}}(|0\rangle^{\otimes n} + |1\rangle^{\otimes n})$ to the interaction dynamics which is encoded by an unknown parameter θ. We consider the interaction dynamics represented by the time evolution $U(\theta) = e^{-iH\theta}$ with the Zeeman Hamiltonian $H = \frac{1}{2}\sum_{i=1}^{n} Z_i$. Then given the global X observable $O_n = X^{\otimes n}$ and the measurement shot number N, the parameter θ can be estimated via

$$\theta^{\text{est}}(n, N) = \frac{\arccos \langle O_n \rangle_\theta}{n}, \tag{22}$$

where $\langle O_n \rangle_\theta$ refers to the expectation value of O_n. The circuit we utilized includes the GHZ state preparation circuit and the free evolution circuit parameterized by θ, as depicted in Fig. S2 **b**. We train the models with circuit parameterized by randomly sampled angles within a continuous interval from 0.1 to 5.0 with increments of 0.1, performing 10 independent experimental

TABLE VI. Dataset characteristics

Task	Training Set		Test Set		Quantum System Size
	sequences	hard regime	sequences	steps	
QAOA-type circuits	50-500	11-20	200	1-20	6
Qubit-wise circuits for quantum metrology	100-500	5-10	500	1-10	1-10

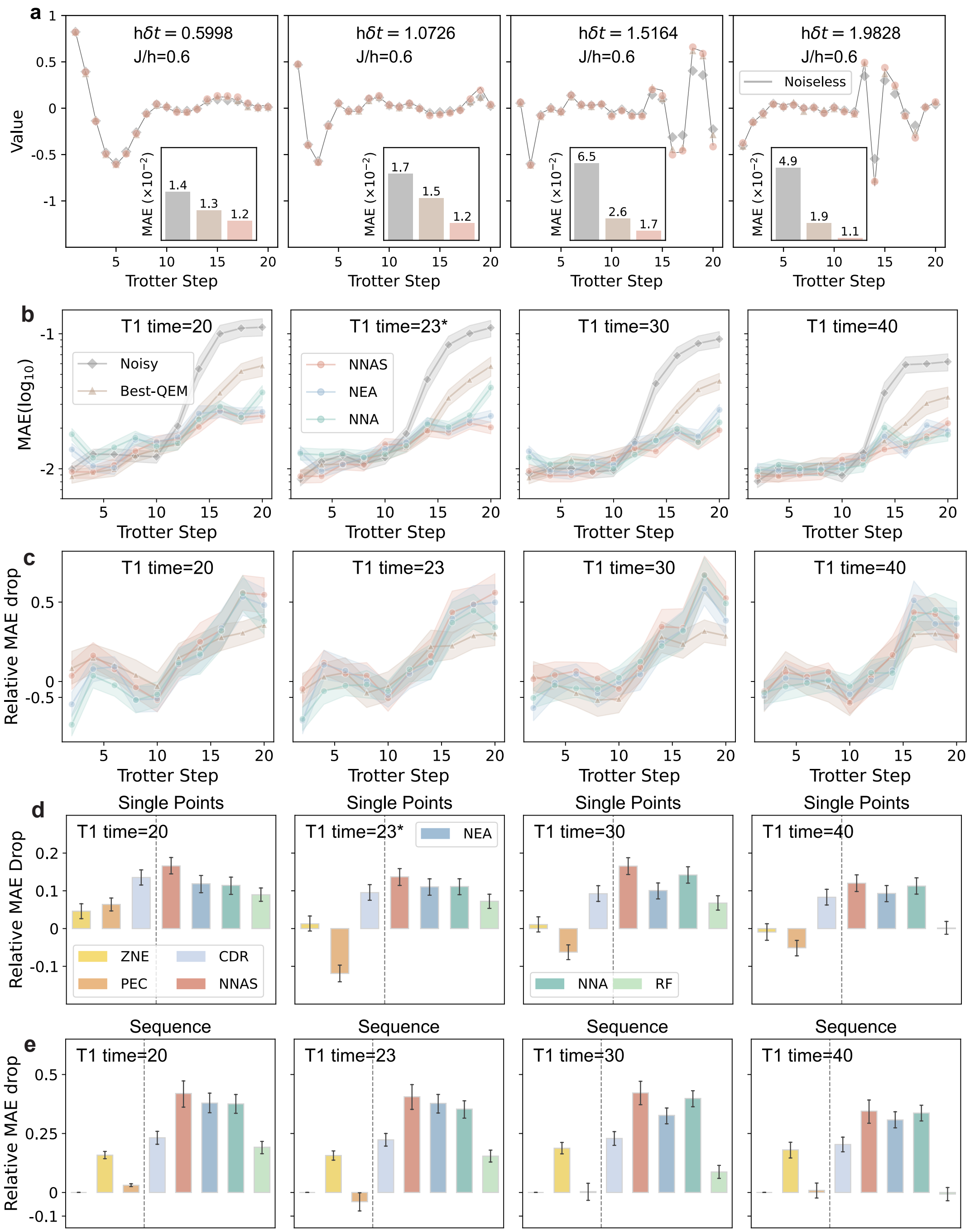

FIG. S3. **Extended data on accuracy analysis for Quantum Approximate Optimization Algorithm-like circuits.** **a** The Trotter evolution for typical values of circuit parameter $h\delta t$ within $[0.5, 2]$. The inset figure shows the total MAE comparison for Trotter steps 1-20 among Noisy, Best-QEM (the best-performing QEM method among standard QEM methods and RF), and our model. **b-c** The comparative results of MAE and the relative MAE drop (RD) for baselines and our model with varying Trotter steps, where MAE and RD are both calculated using single points. These results are derived under diverse noise levels, characterized by the T1 relaxation time. A higher T1 time signifies a lower noise level. **d-e** The comparative analysis of RD for baselines calculated using single points and sequence norms. All the data in the figure presents average results with 95% confidence interval error bars.

trials to estimate metrics for each parameter in simulations. Evaluations are based on angles taken from 0.05 to 5.05 at 0.1 increments.

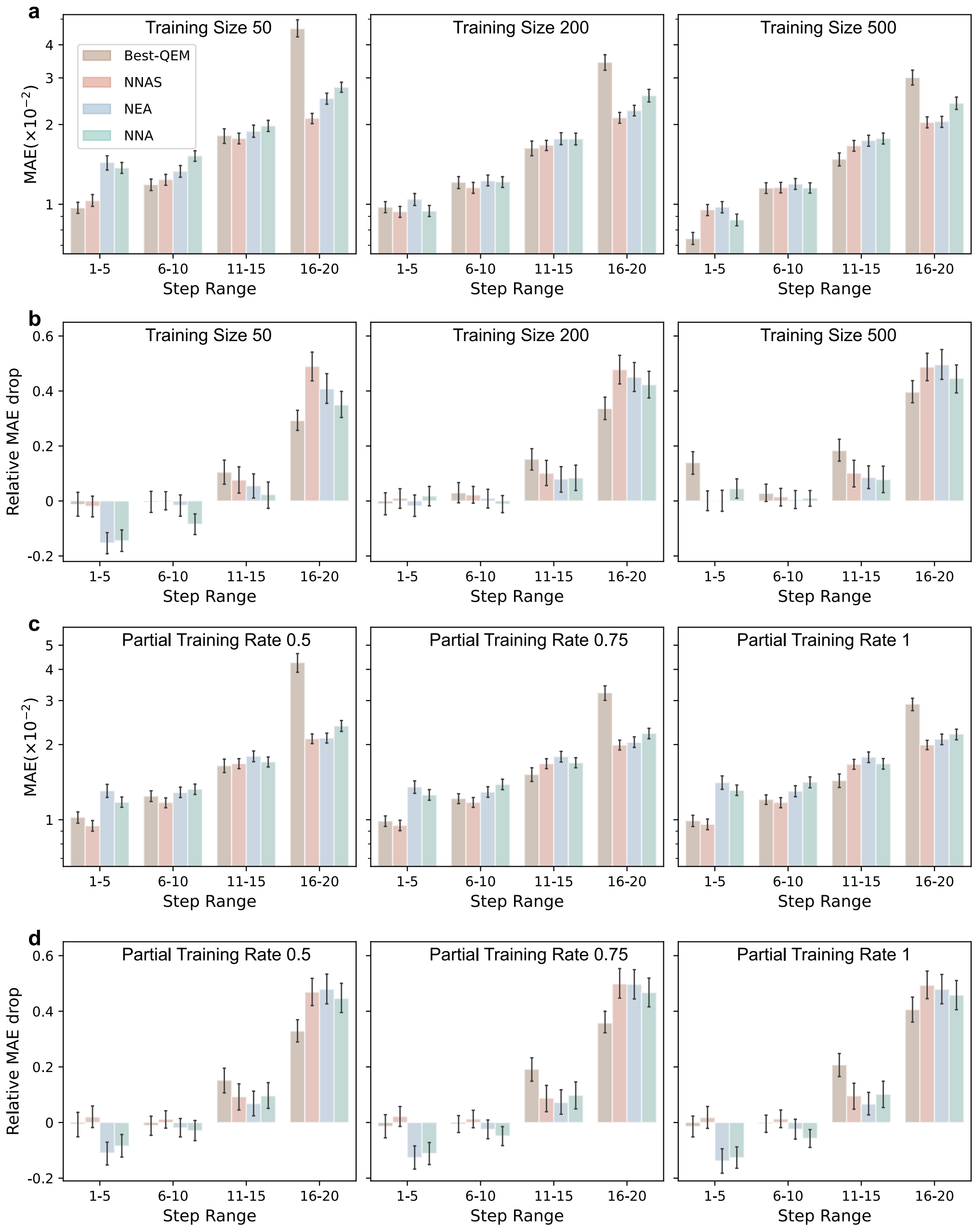

FIG. S4. **Extended data on efficiency analysis of ML-QEM models for Quantum Approximate Optimization Algorithm-like circuits.** The average performance for MAE and RD is assessed with varying training set size and partial training rate across specific Trotter step ranges: 1-5, 6-10, 11-15, and 16-20 (The model's performance with a training set size 100 and a partial training rate 0.25 is discussed in the main text.) All the data in the figure presents average results with 95% confidence interval error bars.

E. Summary of Dataset characteristics

In conclusion, we summarize the dataset utilized in our experiments for two type of circuits. The overall characteristics of the dataset are given in Table VI. For the first two sequential tasks in our simulation experiments, the dataset comprises sequences with varying lengths in the training set, which are randomly truncated, and sequences with fixed lengths in the test set.

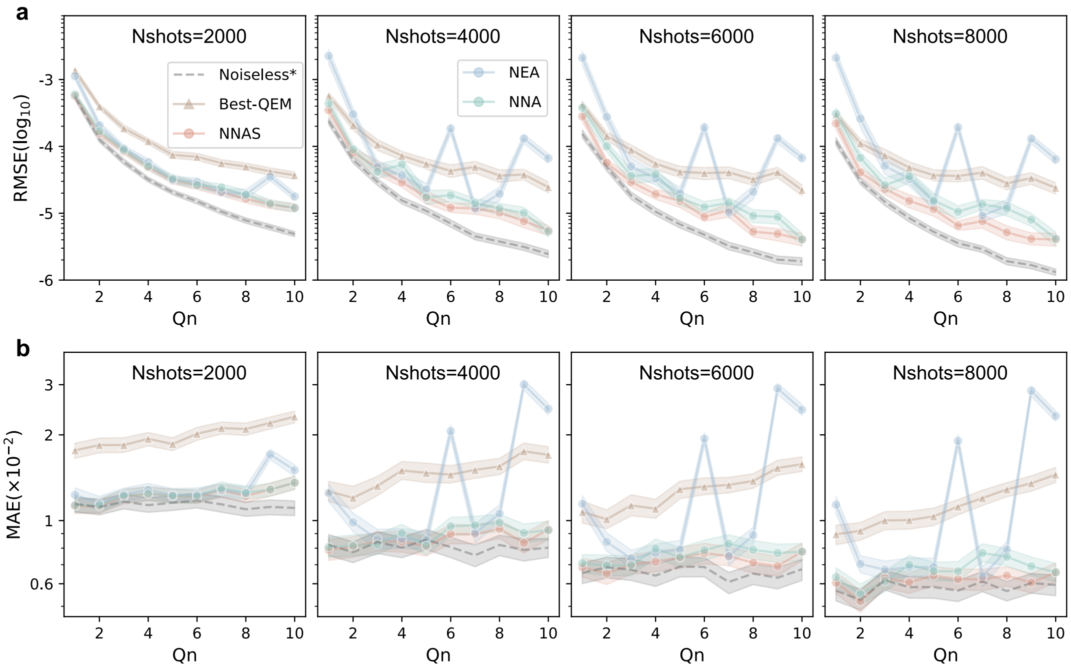

FIG. S5. **Extended data on accuracy analysis for quantum metrology based on GHZ state. a-b** The average performance of the baselines is assessed using two metrics including RMSE for estimated angles and MAE for mitigated expectation values, at each qubit number across an increasing range of measurement shots. Here, the noiseless case refers to the condition that the results only suffer measurement statistical error due to finite measurement shots number. All the data in the figure presents average results with 95% confidence interval error bars.

IV. EXPERIMENTAL DETAILS AND EXTENDED RESULTS

In this section, we detail the implementation of our experiments and present extended results.

A. Experiments for QAOA-type circuits

For layer-wise QAOA-type circuits, we initially select representative results from the interval $[0.5, 2]$ of the circuit parameter $h\delta t$ to demonstrate the noiseless expectation values of the Trotter evolution. As observed in Fig. S3 **a**, the absolute value of these noiseless expectation values increases with the $h\delta t$ value, particularly at higher layer depths, where our model's superiority becomes increasingly evident.

In the main text, we assessed our model's performance using Mean Absolute Error (MAE), a key metric for quantifying the precision of QEM methods. Results as given in Fig. S3 **b** substantiate the superiority of our model in two pivotal aspects. The first is that, as analyzed in the main text, our model demonstrates superior performance with larger Trotter steps. The second is that our model exhibits enhanced stability relative to our ablation models, notably at lower Trotter steps where the ablation models exhibit instability. This stability underscores the robustness of our model's design, which sustains effective mitigation ability as validated by ablation studies.

We then extend our analysis with an additional metric to further investigate the model's performance. We define the relative MAE reduction (RD) as a metric to quantify the relative performance improvement of mitigation methods over the noisy baseline, assessing this for each layer individually. The RD is calculated using the logarithmic ratio of the MAE for the noisy and mitigated expectation values:

$$\mathrm{RD} = \log\left(\frac{|\tilde{y}_l - y_l|}{|y_l^{\mathrm{em}} - y_l|}\right), \tag{23}$$

where $\tilde{y}_l$ denotes the noisy expectation value at l-th layer, y_l represents the noiseless expectation value, and y_l^{em} is the mitigated result obtained from QEM methods. Notably, a larger RD signifies better performance, indicating a more effective mitigation of noise. Fig. S3 **c** depicts the RD of baselines as a function of Trotter Step, where our model shows an overall stable advantage over the baselines. The comparative analysis of RD for baselines under different noise levels is given in Fig. S3 **d** and **e**, using two calculations for single points and sequences.

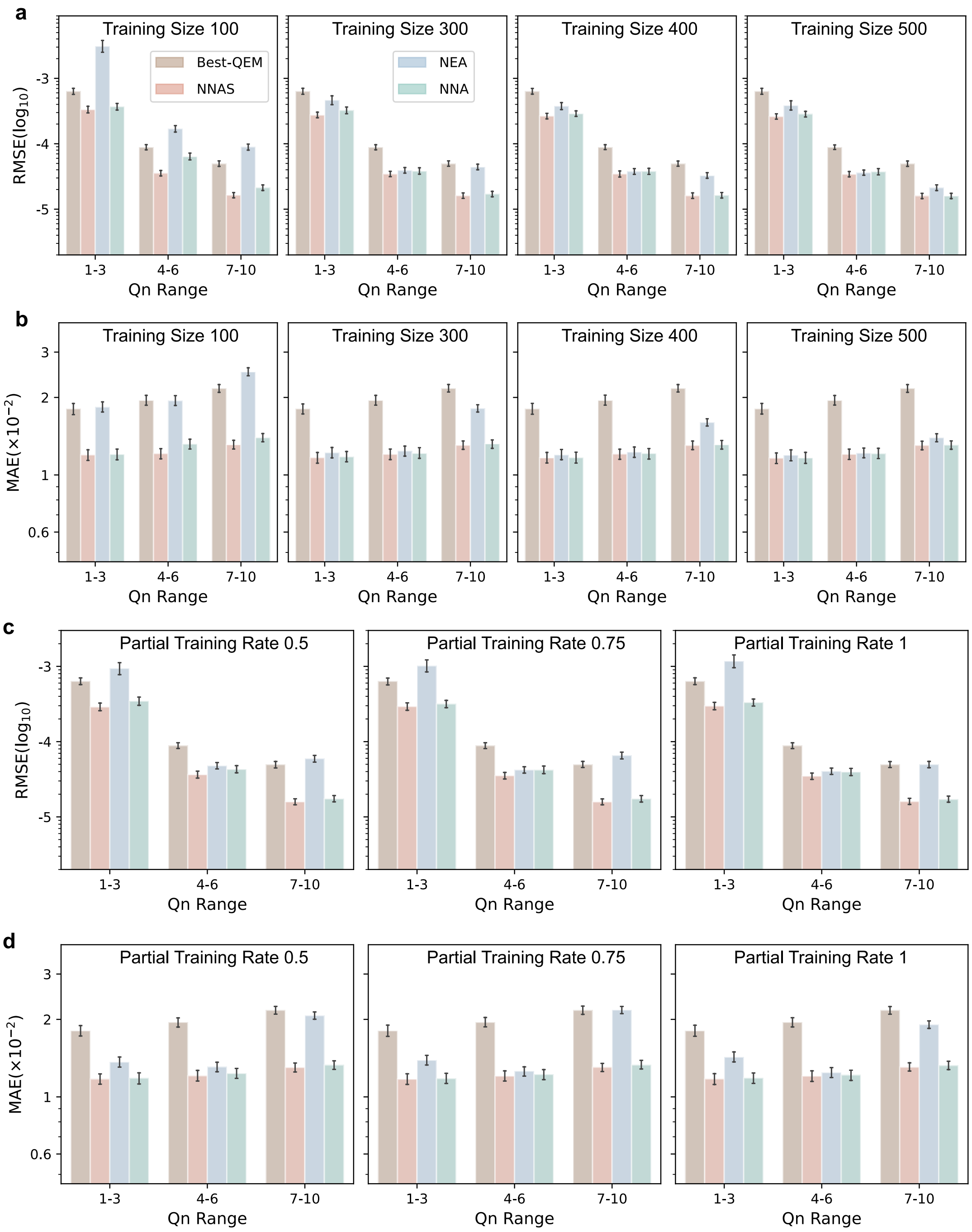

FIG. S6. **Extended data on efficiency analysis for quantum metrology based on GHZ state.** The average performance of RMSE for estimated angles and MAE for mitigated expectation values is assessed with varying training set size and partial training rate across specific qubit number ranges: 1-3, 4-6, and 7-10 (The model's performance with a training set size 200 and a partial training rate 0.25 is discussed in the main text.) All the data in the figure presents average results with 95% confidence interval error bars.

We also evaluate the efficiency of our model in comparison with baselines using ML models, including RF, and our ablated models. As depicted in Fig. S4, the average performance for MAE and RD is assessed across specific Trotter step ranges: 1-5, 6-10, 11-15, and 16-20. This assessment is conducted for two different conditions: training set sizes ranging from 50 to 500 as shown in panels **a** and **b**, and partial training rates varying from 0.5 to 1 as illustrated in panels **c** and **d** (The model's performance with a training set size 100 and a partial training rate 0.25 is discussed in the main text.). Our model outperforms RF and ablation models, particularly with lower training set sizes and partial training rates, demonstrating its strength in small data scenarios.

B. Discussion on Accumulation Surrogating

We further explore how our physical-inspired network surrogates the noise accumulation process. For QAOA-type circuits with up to 12 Trotter steps, we examine a specific instance, visualizing both the quantum cumulative noise $\mathcal{N}$ and the corresponding surrogate $\hat{\mathcal{N}}$ derived from the neural network. Our visualization technique employs the Pauli Transfer Matrix (PTM) representation of $\mathcal{N}$, condensed through sum pooling, alongside the computation of the outer product $\hat{\mathcal{N}}^T\hat{\mathcal{N}}$ to represent $\hat{\mathcal{N}}$. We proceed to conduct a quantitative analysis, expanding our case study to encompass a statistical analysis that includes 200 sequences. This comprehensive approach, combining both visualization and quantification, substantiates our model's rapid detection of structural changes and its structural alignment with quantum cumulative noise. Next, we will delve into the computational details that underpin these findings.

In our experiments for six-qubit QAOA-type circuits, we confront the challenge of visualizing the immense PTM size of (4096,4006). To address this, we utilize sum pooling, which not only simplifies the data representation by reducing dimensionality but also retains critical information. We employ the sum pooling on a sub matrix of size 128, and the resulting size is (32,32). For the surrogate $\hat{\mathcal{N}}$ with an experimental size of 32, the outer product results in a square matrix of size (32,32).

When it comes to the correlation analysis in our experiments with six-qubit QAOA-type circuits, we utilize submatrices extracted from the full PTM of $\mathcal{N}$, perform a flattening operation, and then normalize each vector before computing the Spearman correlation coefficient with the flattened and normalized vector of $\hat{\mathcal{N}}^T\hat{\mathcal{N}}$. The normalization step ensures that each feature contributes equally to the correlation analysis, mitigating the impact of scale differences. The Spearman coefficient, denoted as c_{spearman}, is calculated by first ranking the elements of the datasets X and Y after normalization, and then applying the Pearson correlation formula to these ranks. The formula for c_{spearman} is given by:

$$c_{\text{spearman}} = \frac{\sum_i (rg(x_i) - r\bar{g}_x)(rg(y_i) - r\bar{g}_y)}{\sqrt{\sum_i (rg(x_i) - r\bar{g}_x)^2 \sum_i (rg(y_i) - r\bar{g}_y)^2}} \tag{24}$$

where $rg(x_i)$ and $rg(y_i)$ represent the rank values of x_i and y_i after normalization, respectively, and $r\bar{g}_x$ and $r\bar{g}_y$ are the mean rank values of X and Y. This results in a correlation matrix with dimensions of (128,128).

Furthermore, going disordered prompts us to characterize the structural correspondence through differential entropy. We analyze the variation curves of vector $\hat{\mathcal{N}}$ and the flattened vector from the PTM of $\mathcal{N}$, with a length of 4096^2, as the Trotter step increases. The differential entropy is calculated using the formula:

$$h(X) = -\int_{-\infty}^{\infty} f(x) \log f(x)\, dx \tag{25}$$

where $f(x)$ is the probability density function of the continuous random variable X. For these vectors, we utilize the `scipy.stats.differential_entropy` function with the 'auto' method, which selects an appropriate numerical approach based on the sample size. Given the lengths of our vectors, the function is likely to employ the Ebrahimi method for the shorter vector $\hat{\mathcal{N}}$ and the Vasicek method for the longer flattened PTM vector.

C. Experiments for quantum metrology based on GHZ probe state

We employ three metrics to assess the mitigation effectiveness of models for QM, which we will describe in detail.

- Root Mean Square Error (RMSE). The RMSE is commonly used to describe the sensitivity of $\theta^{\text{est}}(n, N)$ to the statistical error given measurement shots number N. The defination of RMSE is

$$\left(\Delta\theta^{\text{RMSE}}(n, N)\right)^2 = \text{Var}[\theta^{\text{est}}(n, N)] = \mathbb{E}[|\theta^{\text{est}}(n, N) - \theta|^2]. \tag{26}$$

  In the noiseless case, $\Delta\theta^{\text{RMSE}}(n, N)$ scales at $\mathcal{O}(n^{-2})$ for all θ, which reaches the Heisenberg scaling.

- MAE of the expectation value $\langle O_n \rangle_\theta$. This metric provides a direct assessment of the efficacy of the mitigation strategies.

- Fitting rate r of the RMSE scaling curve with respect to the qubit number n. We model the RMSE scaling curve using the function b_0/n^r, where b_0 is determined by fitting the RMSE scaling curve of the noiseless case with the function b_0/n^2.

The RMSE and fitting rate serve as indicators for QM, with the latter being an upper-bounded metric where values closer to 2 signify better performance, as they approach the noiseless scenario. The MAE, alongside the RMSE, quantifies the error in our estimates. Since the fitting rate r is calculated by fitting the data across all the qubits, we only illustrate the detailed results for RMSE and MAE at each number of qubits under varying measurement shots numbers, as given in Fig. S5. Here, the noiseless case refers to the condition that the results only suffer measurement statistical error due to finite measurement shots number. Our model demonstrates performance improvements generally in accordance with the noiseless case as measurement shots number increases. This correlation arises from the decreased measurement statistical error. In terms of efficiency analysis as depicted in Fig. S6, the performance of our model aligns with the outcomes observed in prior QAOA-type circuits. This refines the conclusion of the superior effectiveness and efficiency of our model compared to standard QEM and universal ML-QEM, even for qubit-wise quantum algorithm. Notably, in this experiment, the dataset we utilize encompasses a small range of circuit parameters $\{\theta\}$ (at most 50). Relative to universal ML models that rely on directly capturing the relationship between noisy and noiseless results, learning the corresponding relationship from such a dataset is particularly challenging. This further highlights the utility of physical prior knowledge in ML-QEM under conditions of limited training set.

V. TRAINING SET GENERATION OF LARGE-SCALE QUANTUM SYSTEMS

As we have discussed in the main text, the acquisition of training labels (noiseless expectation values) for large-scale quantum circuit implementations presents significant computational challenges, primarily due to the reliance on resource-intensive classical simulations. In this section, we explore two potential solutions to mitigate these challenges. The first approach utilizes near-Clifford circuits to construct the training set, capitalizing on their efficient simulatability. The second approach employs the Low-weight Pauli propagation algorithm [14], which efficiently computes expectation values for low-weight Pauli observables in shallow circuits. Both methods aim to reduce the computational burden and enhance the scalability of training label acquisition for quantum error mitigation.

A. Training with near-Clifford circuits

Similar to the CDR error mitigation method, we use these circuits for training due to their efficient classical simulation, while testing the model on more general circuits. Through experiments on representative quantum parameterized circuits, we demonstrate that our NNAS model can effectively adapt to the patterns in near-Clifford training sets.

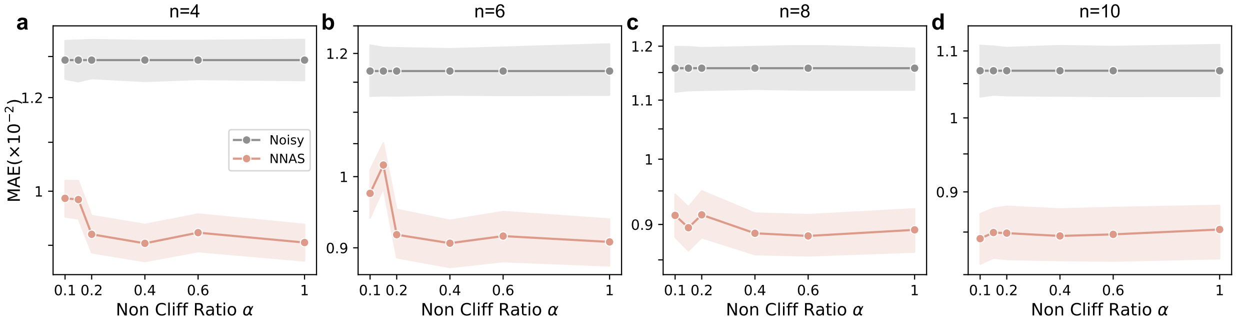

FIG. S7. **Extended data on training with the near-Clifford circuits with varying qubit number and the retention ratio of non-Clifford gates.** The average MAE performance for mitigated expectation values is evaluated using near-Clifford circuits with varying retention ratios α of non-Clifford gates. The near-Clifford training set comprises 500 samples, while the test set, utilizing random parameters with a default $\alpha = 1$, contains 200 samples. All the data in the figure presents average results with 95% confidence interval error bars.

In our simulations, We ultilize an four-qubit hardware-efficient (HE) [15] ansatz circuits, as shown in Fig S2 **c**. A single ansatz layer consists of an alternating trainable single-qubit layers with rotation-y gates, and a fixed entangling layers which is the fully-connected CX gate layer. For a HE circuit with L layers, the ansatz layer is repeated L times, with each layer having independent rotation angles. However, the HE circuit differs from other quantum circuits. Unlike previous tasks with consistent circuit parameters that form a sequence of expectation values across layers/qubits, the HE circuits commonly generate unique outputs per layer depth, lacking a strict sequential pattern. Yet, as noted in the main text, training data can still be recursively collected by layer. When we need to predict the mitigated results for circuits with a fixed layer depth, we can treat it as a sequential and focus solely on the output at the target step.

To better enable NNAS to capture the characteristic information of HE circuits, we have modified the embedding module of NNAS. The embedding methods for various types of features are the same as those mentioned in the maintext except the embedding of rotation angles. We adopt segmented encoding for rotation angles, which divides angles into three categories based on their numerical values and utilizes learnable parameters E_i, where $i \in \{0, 1, 2\}$, for feature representation. Formally, for angle θ, the embedding $f(\theta)$ is defined as

$$f(\theta) = \begin{cases} E_0, & if\ \theta \in [0, \frac{2\pi}{3}) \\ E_1, & if\ \theta \in [\frac{\pi}{3}, \frac{4\pi}{3}) \\ E_2, & \text{otherwise} \end{cases} . \tag{27}$$

Our method of generating near-Clifford HE circuits aligns with Eq. 10, where we vary the retention ratio of non-Clifford gates, which is denoted as α, from 0.1 to 1. We train our model on near-Clifford HE circuits across various α values, using a fixed σ_z^1 observable. The training set comprises 500 samples, while the test set consisted of 200 HE circuits with random parameters. Our experiments cover quantum systems with qubit number n ranging from 4 to 10 and a maximum layer depth of 10. As shown in Fig. S7, the performance of NNAS exhibits only slight fluctuations with varying α, with the impact becoming increasingly minimal as the n grows larger. This robustness can be largely attributed to our segmented encoding, which renders NNAS relatively insensitive to parameter variations. In practical applications, when aiming to efficiently simulate near-Clifford circuits, the maximum permissible number of non-Clifford gates is significantly lower than the order of hundreds [16]. In our experiments, to achieve a precision comparable to that of the original training set ($\alpha = 1$), the minimum number of non-Clifford gates required was found to be less than 30, thereby underscoring the potential of training with near-Clifford circuits.

B. Training label acquisition using low-weight Pauli propagation

Pauli propagation methods are widely utilized for classically simulating expectation values of locally scrambling circuits [17, 18]. These methods operate in the Heisenberg picture by back-propagating Pauli operators through the circuit, where Clifford gates transform Paulis to other Paulis, and non-Clifford gates typically convert a Pauli to a weighted sum of multiple Pauli operators. We employ the low-weight Pauli propagation (LWPP) [14] to efficiently construct training sets for large-scale quantum systems by limiting the number of Pauli terms at each circuit layer.

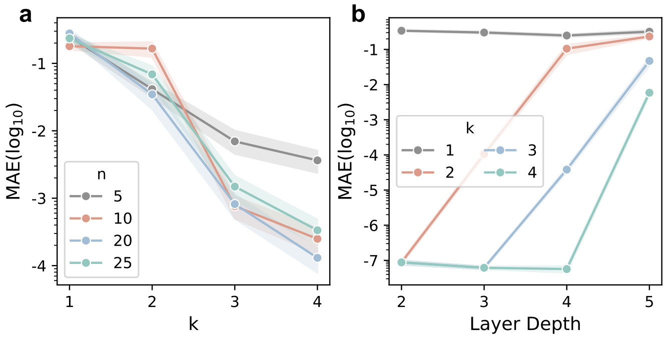

FIG. S8. **Extended data on classical simulation for acquiring training labels via low-weight Pauli propagation (LWPP).** The MAE of simulated values from LWPP is evaluated by comparing with exact values simulated using the `qsimcirq` library. **a** The MAE of LWPP with varying limited number k, with different qubit numbers $n = 5, 10, 20, 25$, averaged across all layer depths of the Hamiltonian evolution (HE) circuit from 2 to 5. **b** The MAE of LWPP with varying layer depth and limited number k. All the data in the figure presents average results with 95% confidence interval error bars.

Suppose the limited number is $k < n$, where n is the qubit number. Given any low-weight observable $O = \sum_{|\sigma_P| \leq k} a_P \sigma_P$, we denote $|\sigma_P|$ as the *weight* (i.e. the number of non-identity terms of σ_P) of the n-qubit Pauli operator σ_P. For j-th ($j = L, L-1, ..., 1$) layer of the quantum circuit with L layers $U = U_1 U_2 ... U_L$, we compute the k-weight approximation of O_{j-1} the using the Heisenberg evolved observable $U_j^\dagger O_j U_j$:

$$O_{j-1} = \frac{1}{2^n} \sum_{|\sigma_P| \leq k} \text{Tr}[U_j^\dagger O_j U_j \sigma_P] \sigma_P. \tag{28}$$

We finally obtain the k-truncated observable $O_U^{(k)} = U_1^\dagger O_1 U_1$ and compute the expectation value by $\tilde{f}_U^{(k)}(O)$ =Tr$[O_U^{(k)} \rho_0]$ where ρ_0 is the initial quantum state. The time complexity of LWPP is $Ln^{\mathcal{O}(\log(\epsilon^{-1}\delta^{-1}))}$ to generate a outcome $\tilde{f}_U^{(k)}(O)$ such that

$$|\tilde{f}_U^{(k)} - f_U(O)| \leq \epsilon ||O||_{\text{Pauli},2}, \tag{29}$$

with probability at least $1-\delta$. Here $f_U(O)$ is the exact value, and we have

$$||O||^2_{\text{Pauli},2} = \sum_{\sigma_P} a_P^2. \tag{30}$$

This demonstrates that achieving any small constant error and failure probability necessitates merely polynomial resources. Typically, when polynomially small error and failure probability are required, the complexity grows quasi-polynomially.

We construct the training set by employing the HE circuits with layer depth ranging from 2 to 5, utilizing the observable σ_z^1, for systems with n varying from 5 to 25. For each circuit instance, we perform simulations with the parameter k ranging from 1 to 4. To evaluate the precision of LWPP, we use the `qsimcirq` library [19] to simulate exact values and compare them with LWPP results, thereby assessing LWPP's performance and precision. Results are given in Fig. S8, where we explore the performance of LWPP with different n,k and layer depth. As shown in Fig. S8 **a**, the MAE of the LWPP decreases very rapidly as k increases. When k=4, the precision has already reached close to 10^{-4} on average across all layer depths. This indicates that LWPP is quite efficient in simulating shallow circuits and is not significantly affected by the increase in the qubit numbers. However, as shown in sub-figure **b**, a limitation of LWPP is that the simulation precision decreases with increasing layer depth. To improve the precision for large depths, it is necessary to increase the limited number k, which in turn increases the computational overhead of the classical simulation. It is worth noting that for lower layer depths, the performance of LWPP is particularly remarkable. When the layer depth is 2, LWPP with $k \geq 2$ can even achieve a precision of 10^{-7}, which is highly consistent with the mechanism of our NNAS model. This is because the recursive design of NNAS allows the training set to contain only a small number of circuits with larger depths, yet still achieve good results on the test set, as demonstrated in the main text.

[1] J. Emerson, M. Silva, O. Moussa, C. Ryan, M. Laforest, J. Baugh, D. G. Cory, and R. Laflamme, Science **317**, 1893 (2007).

[2] E. Magesan, J. M. Gambetta, B. R. Johnson, C. A. Ryan, J. M. Chow, S. T. Merkel, M. P. Da Silva, G. A. Keefe, M. B. Rothwell, T. A. Ohki, *et al.*, Phys. Rev. Lett. **109**, 080505 (2012).

[3] S. Endo, S. C. Benjamin, and Y. Li, Phys. Rev. X **8**, 031027 (2018).

[4] E. van den Berg, Z. Minev, and A. e. a. Kandala, Nat. Phys. **19**, 1116 (2023).

[5] J. E. Jaloveckas, M. T. P. Nguyen, L. Palackal, J. M. Lorenz, and H. Ehm, arXiv:2311.11639 (2023).

[6] H. Liao, D. S. Wang, I. Sitdikov, C. Salcedo, A. Seif, and Z. K. Minev, arXiv:2309.17368 (2023).

[7] Q. Contributors, "Qiskit: An open-source quantum computing software," `https://qiskit.org` (Accessed: 2024-07-11).

[8] J. Ghosh, A. G. Fowler, and M. R. Geller, Phys. Rev. A **86**, 062318 (2012).

[9] Y. Tomita and K. M. Svore, Phys. Rev. A **90**, 062320 (2014).

[10] J. J. Wallman and J. Emerson, Phys. Rev. A **94**, 052325 (2016).

[11] A. Erhard, J. J. Wallman, L. Postler, M. Meth, R. Stricker, E. A. Martinez, P. Schindler, T. Monz, J. Emerson, and R. Blatt, Nat. Commun. **10**, 5347 (2019).

[12] A. Hashim, R. K. Naik, A. Morvan, J.-L. Ville, B. Mitchell, J. M. Kreikebaum, M. Davis, E. Smith, C. Iancu, K. P. O'Brien, I. Hincks, J. J. Wallman, J. Emerson, and I. Siddiqi, Phys. Rev. X **11**, 041039 (2021).

[13] A. Morvan *et al.*, Nature **634**, 328 (2024).

[14] A. Angrisani, A. Schmidhuber, M. S. Rudolph, *et al.*, arXiv:2409.01706 (2024).

[15] A. Kandala, A. Mezzacapo, and others., Nature **549**, 242 (2017).

[16] P. Czarnik, A. Arrasmith, P. J. Coles, and L. Cincio, Quantum **5**, 592 (2021).

[17] T. Begušić, K. Hejazi, and G. K.-L. Chan, arXiv:2306.04797 (2023).

[18] N. A. Nemkov, E. O. Kiktenko, and A. K. Fedorov, Phys. Rev. A **108**, 032406 (2023).

[19] G. Q. AI, "qsimcirq: Cirq interface for qsim," (2023).