Supplementary Information

Gyungmin Cho[1]* and Dohun Kim[1]*

[1]*Department of Physics and Astronomy, and Institute of Applied Physics, Seoul National University, Seoul 08826, Korea*

**Corresponding author: km950501@snu.ac.kr, dohunkim@snu.ac.kr*

Supplementary Information (SI)

Contents

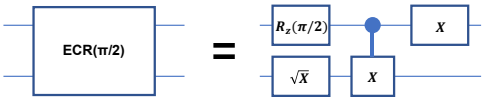
Supplementary Note 1: Device characteristics and a qubit selection

We conducted experiments using fixed-frequency superconducting transmon qubits available through IBM cloud and employed a set of gates {$R_Z(\theta)$ ($\theta \in [0, 2\pi]$), SX (= $R_X(\pi/2)$ up to global phase), X (= $R_X(\pi)$ up to global phase)} to implement arbitrary single qubit gates. The native two qubit gate that can be implemented in the IBM hardware is the cross-resonance (CR) gate, and in the experiment, we used echoed cross-resonance (ECR) pulses[1] to reduce gate errors. We utilized the pre-calibrated ECR($\pi/2$) as the basic building block for the two qubit gate. By

combining ECR(π/2) and single qubit gates, we can implement CNOT gates (Supplementary Figure 1), allowing us to create universal gates along with arbitrary single qubit gates. The metrics for each qubit and gate were summarized in Supplementary Table 1.

T_1	277.91 us
T_2	166.15 us
$\sqrt{X}(=\mathrm{SX})$	2.262×10^{-4}
X	2.262×10^{-4}
ECR(π/2)	7.476×10^{-3}
Readout error	1.090×10^{-2}

Supplementary Table 1 | *ibm_sherbrooke*. Median of the T_1, T_2, gates (SX, X, ECR(π/2)) error and readout error.



Supplementary Figure 1 | ECR(π/2) gate decompositions into hardware native gate set.

In the experiment, we chose a qubit layout with the minimum errors by going through a selection process. Due to the constraints in qubit connectivity, we first identified possible layouts capable of running the given circuit. Next, we used device information about the gates and qubits to estimate the total error for each possible layout. Following this, we conducted the actual experiments using the qubit layout that had the lowest error estimate. Since the device characteristics tend to drift over time, we ran experiments on the layout predicted to perform best at the time of the experiment, and the qubits employed in each experiment can be found in Supplementary Note 5. We used the software package mapomatic[2] and Qiskit[3] for this task.

Supplementary Note 2: Classical shadow and its applications

2.1. Classical shadow

To convert the quantum state into a classical form, quantum state tomography (QST) has been used. For QST[4], Pauli basis measurements were usually employed, which have been executed on a variety of quantum hardware platforms[5,6]. Although generalized measurements or even coherent measurements on multiple copies still have an exponential scaling in sample complexity for QST, entire information of the state ρ may not be necessary if we are interested in the expectation value of a local operator O. Inspired by this, a state learning method called classical shadow was introduced and implemented experimentally[7,8]. To obtain the classical shadow of a quantum state, we apply a unitary transformation sampled from a specific random unitary ensemble to the state, and then measure the resulting state. In our experiment, we used Haar(2)$^{\otimes n}$ as the random unitary ensemble. For example, after performing T samplings of random unitary and measurements, we can obtain the experimental results $S_T(\rho)=\{b^{(t)}, U^{(t)}\}_{t=1}^{T}, (b^{(t)} \in \{0,1\}^n, U^{(t)}=\otimes_{i=1}^{n} U_i^{(t)})$ where $U_i^{(t)}$ is sampled from the Haar measure over $\mathbb{U}(2)$, from which the empirical average for the quantum state ρ can be written as follows

$$\hat{\sigma}_T(\rho)=1/T\sum_{t=1}^{T}\hat{\rho}_t$$

$$\text{where}\quad \hat{\rho}_t=\otimes_{i=1}^{n}(3U_i^{(t)\dagger}\left|b_i^{(t)}\right\rangle\left\langle b_i^{(t)}\right|U_i^{(t)}-I_2) \tag{1}$$

I_2 is a 2 by 2 identity matrix. Then, the following facts hold.

Fact 1 Unbiased estimator[7]. *Define* $\hat{o}=Tr(O\hat{\rho})$ *and use single-shot estimator* $\hat{\rho}=\otimes_{i=1}^{n}(3U_i^{\dagger}\left|b_i\right\rangle\left\langle b_i\right|U_i-I_2)$ *where each* U_i *is sampled from Unitary 2-design over* $\mathbb{U}(2)$*, then expectation value of* $\hat{o}$ *gives*

$$\mathbb{E}_{U,b}(\hat{o})=\mathrm{Tr}(O\rho) \tag{2}$$

And the associated variance is given by the following.

Fact 2 Upper bound on the variance of estimation[7]. *Define* $\hat{o} = Tr(O\hat{\rho})$ *and use single-shot estimator* $\hat{\rho} = \otimes_{i=1}^{n}(3U_i^{\dagger}|b_i\rangle\langle b_i|U_i - I_2)$ *where each* U_i *is sampled from Unitary 3-design over* $\mathbb{U}(2)$*, then upper bound of the variance of* $\hat{o}$ *obeys*

$$\mathrm{Var}(\hat{o}) \le 2^{\mathrm{locality}(O)} \|O\|_2^2 \le 4^{\mathrm{locality}(O)} \|O\|_\infty^2 \tag{3}$$

where locality(O) is the number of qubits where operator O acts non-trivially. So, using empirical average $\hat{\sigma}_T(\rho)$ of (1) in the main text, we can reduce the variance by increasing the number of experimental trials (T). In experiments, we used Haar(2)$^{\otimes n}$ as a random unitary ensemble, so the above two facts automatically satisfied.

Using the Fact1 and 2, we can prove the following Corollary1.

Corollary1 Virtual gate by post-processing on the classical shadows. *If unitary* V *can be decomposed into tensor products of single qubits gate (*$V = \otimes_{i=1}^{n} V_i$*), Using the classical shadows* $S_T(\rho) = \{b^{(t)}, U^{(t)}\}_{t=1}^{T}$*, we can obtain unbiased estimator* $\hat{o}_V = \mathrm{Tr}(OV\hat{\rho}V^{\dagger})$ *with the same upper bound on the variance as in Fact 2.*

Proof. we can directly use the result from Fact 1, 2.

$$\begin{aligned} \mathbb{E}(\hat{o}_V) &= \mathbb{E}(\mathrm{Tr}(OV\hat{\rho}V^{\dagger})) \\ &= \mathrm{Tr}(OV\mathbb{E}(\hat{\rho})V^{\dagger}) \qquad (\because \text{linearity of } \mathrm{Tr}(\cdot)) \\ &= \mathrm{Tr}(OV\rho V^{\dagger}) \qquad (\because \text{ Fact 1}) \end{aligned} \tag{4}$$

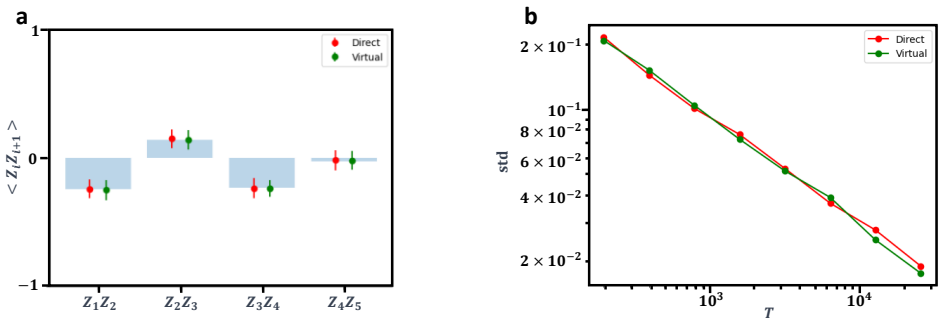
As expected, it gives an unbiased result. And we can obtain upper bound on the variance.

$$\begin{aligned} \mathrm{Var}(\hat{o}_V) &\le 2^{\mathrm{locality}\left(V^{\dagger}OV\right)} \left\|V^{\dagger}OV\right\|_2^2 \quad (\because \mathrm{Tr}(OV\hat{\rho}V^{\dagger}) = \mathrm{Tr}(V^{\dagger}OV\hat{\rho})) \\ &= 2^{\mathrm{locality}(O)} \|O\|_2^2 \end{aligned} \tag{5}$$

In the second line, we use the fact that tensor products of single qubit gates do not change the locality of O and matrix norm $\|\cdot\|_2$ is invariant under the unitary transformation. □

Numerical simulation for virtual gate

We provide a numerical simulation of the virtual gate method for estimating the expectation value for a given quantum state. In our simulation, we prepared a 5-qubit GHZ state with a random unitary applied ($U_{\text{random}}|\text{GHZ}\rangle$ where U_{random} is a tensor product of single qubit gates) and tried to estimate the expectation values of $\{Z_iZ_{i+1}\}_i$ via the classical shadow. As shown in Supplementary Figure 2a, the estimated values obtained from post-processing on classical shadows of GHZ state and the classical shadow from directly applying the random unitary both yield the same unbiased results. Additionally, both methods exhibit the same scaling of standard deviation (std) as the number of experimental trials (T) increases, as depicted in Supplementary Figure 2b. For the circuit simulation, we used the software package Qiskit[3].



Supplementary Figure 2 | Classical simulation of virtual gate method. **a**, blue bars represent the exact values for each observable Z_iZ_{i+1}, the red ones denote the estimated values using the classical shadow obtained through T = 1600 randomized measurements (RM) after actually applying the tensor products of single qubit gates, and the green ones are the values estimated through virtual gate method. The error bar is the standard deviation (std) from 100 data points. **b**, the scaling behavior of standard deviation on the number of randomized measurements (T).

2.2. Shadow kernel

In our quantum phase classification experiment, we employed the shadow kernel (equation (3)) in the main text to learn the non-linear properties of the state ρ. If we choose the kernel function $k_{\text{linear}}(x, x') = \text{Tr}(\rho(x)\rho(x'))$ for the task of distinguishing quantum phase where $\rho(x)$ is the ground state of $H(x)$, then there must exist a function $f(\rho) = \text{Tr}(O\rho)$ that can be used for phase classification, such that $f(\rho) \geq 0$ if $\rho \in$ phase A and $f(\rho) < 0$ if $\rho \in$ phase B. However, recent studies have indicated that using such linear observables is likely to fail when the system is not translational invariant[9,10]. As a result, a more expressive kernel is needed to perform phase classification in more general setting, prompting us to use the shadow kernel for ML. If a function $f(\rho)$ for some non-negative integer r and d_p

$$f(\rho) = \sum_{d=0}^{d_p} 1/d! \sum_{A_1,\ldots A_d \subset \{1,\ldots,n\},\ |A_i| \leq r} \text{Tr}(O_{A_1,\ldots,A_d} \text{Tr}_{\neg A_1}(\rho) \otimes \cdots \otimes \text{Tr}_{\neg A_d}(\rho)) \tag{6}$$

where $\neg A$ is a complement of A exists that can classify the given phases, then the shadow kernel can be used to learn the function $f(\rho)$. Furthermore, if the system's correlation length is independent on the system size, it has been proven that learning $f(\rho)$ can be done using only polynomial sample and computation time[10]. During the ML in our case, we utilized a modified shadow kernel (Supplementary Eq. (7)) when the kernel matrix obtained from the shadow kernel was unstable, rather than the original shadow kernel equation (equation (3)) in the main text.

$$k_{\text{shadow}}(S_T(\rho),\ S_T(\tilde{\rho})) = \exp\{\tau / T(T-1) \sum_{t \neq t'}^{T} \exp[\gamma / n \sum_{i=1}^{n} \text{Tr}(\sigma_i^{(t)} \tilde{\sigma}_i^{(t')})]\}$$

$$\text{where } \sigma_i^{(t)} = 3U_i^{(t)\dagger} \left| b_i^{(t)} \right\rangle \left\langle b_i^{(t)} \right| U_i^{(t)} - I_2 \tag{7}$$

Supplementary Note 3: Various error reducing procedures

We conducted classical ML using data from a quantum computer without quantum error correction, which means that various types of errors exist in the data. If there are a high level of errors in the data, it becomes difficult to obtain accurate predictions from the trained ML model even with using effective ML algorithms. Therefore, to achieve better performance of ML model, it is imperative to obtain data with reduced errors. In this Note, we will introduce various error reducing procedures used in experiments.

3.1. Dynamical Decoupling (DD)

During the experiment in the main text, the gate time for single-qubit and two-qubit gates are different, and there are cases where gates are intentionally not applied to some qubits when running circuits. As a result, while operations are performed on some specific qubits, the remaining qubits are idle, leading to dephasing errors. DD is a technique to counteract these errors by inserting a logical identity sequence during the idle period, causing coherent errors to refocus[11,12]. Among many possible pulse sequences, we utilized [X($+\pi$), Y($+\pi$), X($-\pi$), Y($-\pi$)] during the idle period in the all experiments of main text (In cases where the duration is insufficient to implement the above sequence, [X($+\pi$), X($-\pi$)] is utilized instead).

3.2.Pauli Twirling (PT)

While DD is applied to idle qubits, PT consists of adding a pair of Pauli gates before and after the two-qubit Clifford gates which doesn't change the logical outcome[13]. This allows us to transform some coherent errors in two-qubit gate into random decoherent errors. Given that, in the worst case, coherent errors can accumulate quadratically with circuit depth compared to

decoherent errors, PT can help decrease the errors in the experiment. The native two-qubit gate employed in the experiment is ECR(π/2) which is in the Clifford group. We implemented PT by uniformly sampling a Pauli gate $P \in$ {II, IX…ZY, ZZ} in front of each ECR(π/2) gate and inserting another Pauli gate $P' = \mathrm{ECR}(\pi/2)\cdot P\cdot\mathrm{ECR}(\pi/2)^{\dagger}$ afterward to compensate the logical effect of the P. Because altering the gate sequence for each measurement would be resource-intensive, we used 5 Pauli-twirled circuit instances in each circuit (But, when acquiring classical shadows, PT was implemented for all T randomized measurements).

3.3. Post-selection (PS)

To obtain data with fewer errors, one approach is to reduce the errors themselves. Alternatively, post-processing methods that detect errors and discard the erroneous data can also be used to achieve fewer-error data, and PS falls under this category. For instance, in a second-quantized system with a conserved number of particles, where $|0\rangle$ means the absence of particles and $|1\rangle$ the presence of the particle. After the measurement, If the total number of particles is not conserved, it indicates errors occurred during the state preparation or measurements. By excluding this data during the analysis, results with reduced errors can be acquired at the cost of more measurements than before.

In the experiment of main text, PS was used in estimating site correlation $\langle a_i^{\dagger} a_j \rangle$ for the ground state of a 1D nearest-neighbor random hopping system. The ground state of the given system should be half-filled, so in a system with n sites, only n/2 (n = even) sites was occupied. If measurement outcomes give different particle numbers, we can omit that result. Such an approach can be utilized in broader scenarios where the parity is conserved[12].

3.4. Macweeny-purification

In the experiment of predicting ground state properties, our goal was to learn the correlation matrix C where C_{ij} corresponds to site correlation $\langle a_i^\dagger a_j \rangle$ of the ground state. In cases like the 1D nearest-neighbor random hopping system, where the system's ground state is expressed as a Slater determinant, the correlation matrix becomes idempotent ($C^2 = C$). However, in actual experiments, errors occurred during quantum state preparation or measurements and statistical fluctuation of measurement outcomes cause correlation matrix C_{exp} to deviate from idempotency. Therefore, a process of projecting it onto an idempotent matrix is helpful, which can be implemented using Macweeny-purification[14]. Macweeny-purification involves iteratively repeating the formula (Supplementary Eq. (8)).

$$C_{k+1} = C_k^2(3I - 2C_k) \tag{8}$$

The correlation matrices drawn in Fig. 2 of the main text correspond to the purified version of the raw correlation matrix C_{exp} and C_{ML}. A detailed error analysis of Macweeny-purification can be found in the literature[14].

3.5. Parity measurement via circuit recompiling

We were able to eliminate some experimental results through post-selection (PS), but in order to do this, the circuit must respect a certain symmetry. For example, in the experiment for predicting ground state properties described in the main text, the applied circuit should not change the number of particles to allow post-selection to proceed.

The quantity $a_i^\dagger a_j$ that we want to measure in the experiment is not Hermitian when $i \neq j$. But, $\langle a_i^\dagger a_j \rangle$ can be obtained by combining the measurement results of $\langle a_i^\dagger a_j + a_j^\dagger a_i \rangle$ and $\langle \mathrm{i}(a_i^\dagger a_j - a_j^\dagger a_i) \rangle$ which are Hermitian. However, in our case, the unitary and initial states of the circuit

are all composed of real values, so $\langle a_i^\dagger a_j \rangle = 2 \cdot \mathrm{Re}(\langle a_i^\dagger a_j + a_j^\dagger a_i \rangle)$, and it is transformed into $1/4(X_i Z_{i+1} .. Z_{j-1} X_j + Y_i Z_{i+1} .. Z_{j-1} Y_j)$ by the Jordan Wigner transformation. First, in the case where $i = j$, $a_i^\dagger a_i$ is transformed into $1/2(1 - Z_i)$ in which computational basis measurement is enough. The second case where $j = i+1$, we need to measure $1/4(X_i X_{i+1} + Y_i Y_{i+1})$, and since $X_i X_{i+1}$ and $Y_i Y_{i+1}$ are commute, we can simultaneously measure them by rotating the measurement basis employing $U_{\mathrm{single}}^{\otimes 2}$ where $U_{\mathrm{single}} = R_X(\pi/2) \cdot R_Y(-\pi/2)$ before the measurement. However, the U_{single} does not preserve the particle number, making post-selection impossible for subsequent measurement results. To circumvent this issue, previous research[12,14] has shown that applying $\mathcal{U}_{\mathrm{p}}^{(\mathrm{layer})}$ (consisting of a layer of two qubit gates preserving the symmetry) instead of U_{single} allows for post-selection in measurement results. We further improved this by absorbing the two-qubit gates in $\mathcal{U}_{\mathrm{p}}^{(\mathrm{layer})}$ into the gates required for the state preparation circuit, without incurring any extra gate overhead. In the following, we will provide a detailed derivation of parity measurement via circuit recompiling. To do that, we use some facts.

Fact 3 Basis transformation of Fermionic creation/annihilation operator[15]. *Consider* $A_K = \sum_{i,j} K_{ij} c_i^\dagger c_j$ *where K is N × N Hermitian matrix and* $\mathbf{c}^\dagger = (c_1^\dagger \ldots c_N^\dagger)^T$. *Then, the following holds*

$$\exp(-iA_K)\mathbf{c}^\dagger \exp(iA_K) = K\mathbf{c}^\dagger \tag{9}$$

Using the Fact 3 we can define $\mathcal{U}(U) = \exp(\sum_{ij} (\log U)_{ij}^T c_i^\dagger c_j)$ where T is transpose, then (Supplementary Eq. (9)) change into the following

$$\mathcal{U}(U)\mathbf{c}^\dagger \mathcal{U}(U)^\dagger = U\mathbf{c}^\dagger \tag{10}$$

Using the (Supplementary Eq. (10)), it becomes clear that $\mathcal{U}(U_1 U_2) = \mathcal{U}(U_2)\mathcal{U}(U_1)$ holds.

Fact 4 Givens decompositions[15]. *$M \times N$ ($M \leq N$) matrix Q satisfying $Q^\dagger Q = P_S$ where P_S is the projector onto suitable subspace S which has dimension M can be decomposed iteratively into $N \times N$ Givens matrices (G) and a $M \times N$ diagonal matrix.*

$$Q = U_{\text{diag}} G_{N_G} \ldots G_1 \text{ where } G_i(\theta) = \begin{pmatrix} 1 & & & & & & \\ & \ldots & & & & & \\ & & 1 & & & & \\ & & & \cos\theta & -\sin\theta & & \\ & & & \sin\theta & \cos\theta & & \\ & & & & & 1 & \\ & & & & & & \ldots \\ & & & & & & & 1 \end{pmatrix} \tag{11}$$

N_G is the total number of Givens matrices. The Givens matrices (G) used in this paper are non-trivial only within two neighboring rows and columns. Keep that in mind, when we compute $\mathcal{G} = \mathcal{U}(G)$, it effectively transforms into a two-qubit gate acting on adjacent qubits (Supplementary Eq. (12)).

$$\mathcal{G} = \mathcal{U}(G(\theta)) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{12}$$

Fact 5 Converting a quadratic Hamiltonian into a non-interacting Hamiltonian[15]. *The quadratic Hamiltonian written as* $H = \sum_{i,j=1}^{N} (K_{ij} - \mu\delta_{ij}) c_i^\dagger c_j$ *where $K = K^\dagger$ and μ is the chemical potential can be converted into* $H = \sum_i \varepsilon_i b_i^\dagger b_i + (c \text{ number})$ *where $\boldsymbol{b}^\dagger = U\boldsymbol{c}^\dagger$. As a result, ground state of H can be written as* $|gs\rangle = \prod_{i,\, \varepsilon_i<0} b_i^\dagger |\text{vac}_b\rangle = \mathcal{U}(U) \prod_{i,\, \varepsilon_i<0} c_i^\dagger |\text{vac}_c\rangle$, $(^\forall i,\ b(c)_i |\text{vac}_{b(c)}\rangle = 0)$.

In experiments for predicting ground state properties, we use the Hamiltonian $H_{\text{hop}} = \sum_i x_i c_i^\dagger c_{i+1} + h.c.$ which is a special case of quadratic Hamiltonian. Utilizing Fact 5, we

can transform H_{hop} into a non-interacting Hamiltonian $H_{\text{hop}} = \sum_i \varepsilon_i b_i^\dagger b_i + (c \text{ number})$ where $\boldsymbol{b}^\dagger = U\mathbf{c}^\dagger$. By keeping i^{th}-row where $\varepsilon_i < 0$, we can obtain the $M \times N$ matrix Q ($Q^\dagger Q = P_S$) from U. Using Fact 3 and 4, by applying the unitary $\mathcal{U}$

$$\begin{aligned} \mathcal{U} &= \mathcal{U}(G_{N_G}...G_1) \\ &= \mathcal{U}(G_1)...\mathcal{U}(G_{N_G}) \end{aligned} \tag{13}$$

to the half-filled $|1...10...0\rangle$ state, we can prepare the ground state up to a global phase.

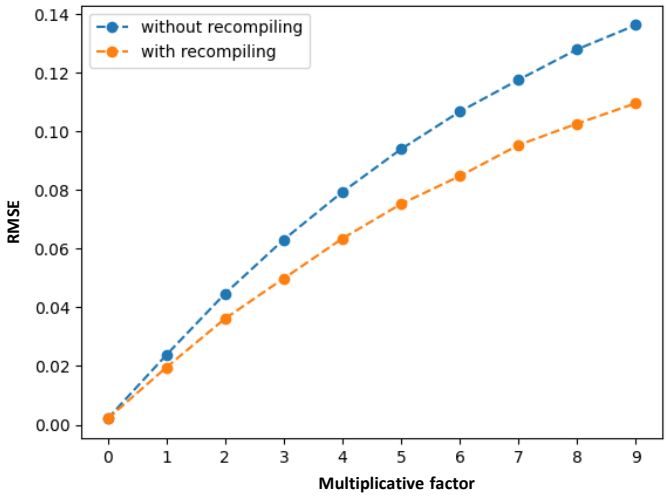
In the experiment, in addition to $\mathcal{U}$ in (Supplementary Eq. (13)), a basis rotation $\mathcal{U}_{\text{p}}^{(\text{layer})}$ is needed for parity measurement. Taking this into account, the total unitary to be implemented through a quantum computer is $\mathcal{U}_{\text{total}} = \mathcal{U}_{\text{p}}^{(\text{layer})} \cdot \mathcal{U}$. Each two-qubit gate in $\mathcal{U}_{\text{p}}^{(\text{layer})}$ can be obtained by diagonalizing $1/4(X_iX_{i+1} + Y_iY_{i+1})$.

$$1/4(X_iX_{i+1} + Y_iY_{i+1}) = U_{\text{p}} D U_{\text{p}}^\dagger \quad \text{where} \quad U_{\text{p}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \; D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & -1/2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{14}$$

By utilizing (Supplementary Eq. (14)), $1/4\,\text{Tr}((X_iX_{i+1} + Y_iY_{i+1})\rho) = \text{Tr}(U_{\text{p}} D U_{\text{p}}^\dagger \rho) = \text{Tr}(D U_{\text{p}}^\dagger \rho U_{\text{p}})$ holds. As a result, $U_{\text{p}}^\dagger$ is adequate for the parity measurement. However, considering that the error of two-qubit gates is an order of magnitude larger than single-qubit gate, additional errors from $U_{\text{p}}^\dagger$ in $\mathcal{U}_{\text{p}}^{(\text{layer})}$ occurs. Yet, taking advantage of $U_{\text{p}}^\dagger = \mathcal{U}(\text{G}(\pi/4))$, we are able to accomplish parity measurement without any additional two-qubit gates through the following circuit recompiling.

$$\begin{aligned} \mathcal{U}_{\text{total}} &= \mathcal{U}_{\text{p}}^{(\text{layer})} \mathcal{U}(G_{N_G}...G_1) \\ &= \mathcal{U}(P)\mathcal{U}(G_1)...\mathcal{U}(G_{N_G}) \\ &= \mathcal{U}(G_{N_G}...G_1 P) \\ &= \mathcal{U}(G'_{N_G}...G'_1) \\ &= \mathcal{U}(G'_1)...\mathcal{U}(G'_{N_G}) \end{aligned} \tag{15}$$

Through this, we achieved the desired $\mathcal{U}_{\text{total}}$ with a new set of Givens rotations $\{G_i'\}$. By adopting the circuit recompiling method outlined above, we can eliminate the errors from parity measurement. In Supplementary Figure 3, we conducted a noisy classical simulation while varying the degree of depolarizing error rate and observed that the total error decreased in all depolarizing strength.



Supplementary Figure 3 | Simulation of 6 qubits random hopping system to determine the degree to which parity measurement via circuit recompiling can reduce errors. The blue points indicate the actual implementation of $\mathcal{U}_{\text{p}}^{(\text{layer})}$, and the orange points depict the outcomes with circuit recompiling. In the noisy simulation, we imposed the depolarizing channel with error rate $(p_{\text{single}}, p_{\text{two}}) = (0.001, 0.01) \times$ (multiplicative factor) on each single-qubit and two-qubit gate.

Until now, we have focused on a specific case where $j = i+1$. However, for a more general case where $i \neq j$, it's possible to transform $a_i^\dagger a_j$, using fermionic swap operations, to the form of $a_k^\dagger a_{k+1}$ without additional gates by adjusting angles in Givens rotations[14].

3.6. Measurement-assisted state preparation method

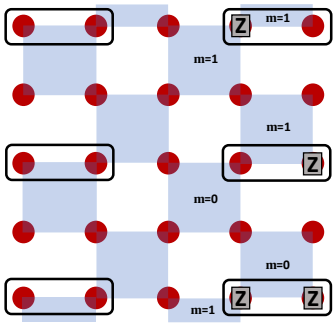
Recently, various methods have been proposed and experimentally realized to prepare topologically ordered states on the quantum computer[16]. However, to implement this, suitable qubit connectivity needs to meet, and if this condition is not satisfied, swap gate overhead will occur. Moreover, even if the qubit connectivity condition was met, to prepare the ground state of the toric code using only unitary gates, a required circuit depth is proportional to the code distance (d_{code}) of toric code.

To overcome such problems, we used the Measurement-assisted state preparation method as a way to detour hardware constraints and reduce errors from additional swap gates[17]. As shown in Fig. 4a in the main text, after applying a series of unitary operations and measuring ancilla qubits, the state after the measurement can be written as follows.

$$|\psi\rangle = 1/\sqrt{2^{N_p}} \prod_p (I + (-1)^{m_p} B_p)|0\rangle^{\otimes n} \tag{16}$$

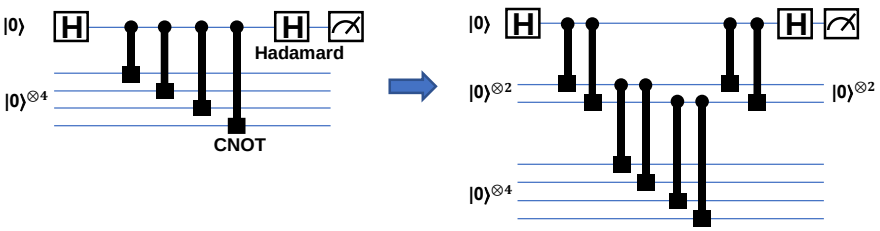
where $B_P = \Pi_{i \in p} X_i$ is a X-plaquette operator, N_p is the number of X-plaquettes and $m_p \in \{0, 1\}$ represents the measurement results of the ancilla qubit present on each X-plaquette.

To create the $|0_L\rangle$ state corresponding to equation (4) in the main text, we need to apply adaptive gates based on the measurement results to ensure all m_p values are 0 in (Supplementary Eq. (16)). In our experiment, by applying the Pauli Z gate depending on the measurement results of the ancilla qubits, we were able to deterministically prepare the $|0_L\rangle$ state. We grouped some of the 25 data qubits as shown in the Supplementary Figure 4 and applied the adaptive Pauli Z gates only within the grouped qubits.



Supplementary Figure 4 | The colored with blue squares corresponds to the *X*-plaquette, and there is a measurement result for each ancilla qubit on the *X*-plaquette. Among the many possible ways, we used the one that applies suitable Z gates to qubits grouped together by black lines.

In Fig. 4a of the main text, 8 CNOT gates were used to realize the projection $\frac{I+(-1)^{m_p}B_p}{2}$. Also, in the heavy-hexagonal lattice, in addition to 1 measurement qubit and 4 data qubits, 2 additional ancilla qubits initialized to $|0\rangle$ are required to apply the CNOT gate from the measurement qubit to the data qubit as in the Supplementary Figure 5.



Supplementary Figure 5 | 4 additional CNOT gate due to the qubit connectivity on heavy-hexagonal lattice

3.7.Virtual gate on classical shadows

During the measurement of ancilla qubits in the Measurement-assisted state preparation method, data qubits remain idle, and considering that the measurement time is much longer than the single and two-qubit gate time in most superconducting devices, unwanted errors may occur during this period, causing the information stored in the qubits to leak out. To reduce the errors from the idle data qubits, we used a virtual gate through post-processing on classical shadows, as explained in the Supplementary Note 2.

To create training data for the topologically ordered phase, we apply a tensor product of random unitary $U = \bigotimes_{i=1} U_i$ (each U_i is sampled independently from Haar(2)) to the fixed-point state $|0_L\rangle$, and perform randomized measurements to obtain the corresponding classical shadows for each state. It is important to note that there are three types of randomness involved here. The first one is for data generation, the second one is the random unitary required to obtain the classical shadow representation of the state, and the last is adaptive gate conditioned on the measurement results of ancilla qubits. In experiments on the quantum hardware, we only applied the random unitary for acquiring classical shadows representation, while the random unitary required for data generation and adaptive gate are done via post-processing on classical shadows.

3.8.Measurement error mitigation (MEM)

During the experiment for Fig. 5 in the main text, we used MEM to correct the measurement errors in the data obtained from the quantum computer. We carried out an additional experiment to construct the response matrix (R) containing information about the correlation of errors that occurred during the measurements. Since we conducted measurements on 4 out of the 9 qubits in the experiment, we prepared possible 16 strings for the 4 qubits and obtained response matrix R through 50,000 measurements for each string. Using the acquired R, we could retrieve the error-mitigated measurement outcome p_{MEM} by employing the inverse relation $p_{\mathrm{MEM}} = R^{-1} p_{\mathrm{exp}}$.

Supplementary Note 4: Comparison with quantum convolutional neural network (QCNN)

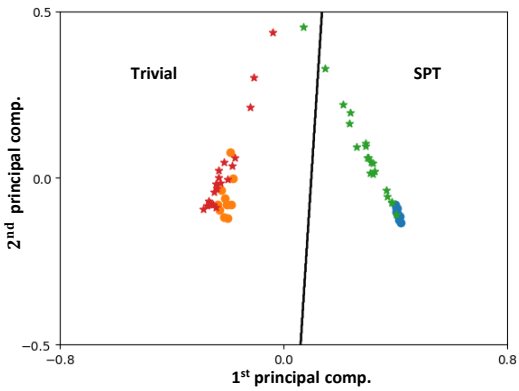
Following the development of QCNN[18], which was inspired by the well-known convolutional neural network (CNN) architecture in classical ML[19], various subsequent studies have been conducted[9,20,21]. There are two main reasons for the success of QCNN. First, while it was inspired by classical ML, the physical roots of QCNN can be found in the well-known multi-scaled entanglement renormalization ansatz (MERA)[22] and tensor renormalization[23] from tensor networks, allowing the ML model to be more than just a simple black box and providing physical interpretability. Secondly, even though many circuit ansätze in quantum machine learning suffer from the barren plateau[24], it is known that QCNN ansatz doesn't[20]. Building on this, experimental realization has shown that even a restricted QCNN compared with original proposal can discern SPT phases with enhanced accuracy than naïve measurements of the string order parameter (SOP) on a 7-qubit superconducting device[21].

Recently, further research has explored the performance of QCNNs in situations without translational invariance[9]. Despite the given state being in the SPT phase, they reported that the performance of the QCNN decreases as the system's non-uniformity increases. QCNNs for distinguishing the SPT phase are based on the SOP and in the process of deriving the SOP, translational invariance of system was assumed[25]. Therefore, in cases without translational invariance, the performance of phase classification through the SOP cannot be guaranteed. Furthermore, in some cases, for linear order parameters $f_{\text{linear}}(\rho) = \text{Tr}(O\rho)$, detecting a specific SPT phase is proven impossible, even O acts non-trivially on the entire system which cannot be decomposed into the sum of local terms[9]. To circumvent this no-go theorem, using $f_{\text{non-linear}}(\rho) = (O\rho^{\otimes k})$ for phase classification is one approach, exemplified by many-body topological invariants (MBTIs)[26].

Experimentally measuring non-linear properties on noisy quantum computers is challenging,

but recent studies have implemented the use of classical shadow or statistical correlation to measure non-linear quantities such as $\mathrm{Tr}(\rho^2)$ and $\mathrm{Tr}(\rho^3)$[27]. In our case, by using classical shadows as data in ML, we demonstrated that phase classification between the SPT phase and the trivial phase is possible through ML, even in cases where the system lacks translational invariance, making the task challenging for SOPs or QCNN to succeed.

In Fig. 3g of the main text, we demonstrated that the ML model trained on states generated from fixed-points state can also be applied to distinguish the phases of translationally invariant systems, such as $H_{\mathrm{CI}} = -J\sum_i Z_i X_{i+1} Z_{i+2} - h_1 \sum_i X_i - h_2 \sum_i X_i X_{i+1}$ used in previous studies[18,21]. To obtain the classical shadow representation of the ground state of the H_{CI}, we employed the DMRG method. After fixing $J = 1$, we performed sampling of h_1 and h_2 from [0.0, 1.6] and [-0.8, 1.6]. By mapping these data points to the space with the phase boundary from the trained ML model, we can classify between SPT and trivial phases with high accuracy. Distributions of data mapped on the suitable space are shown in the Supplementary Figure 6.



Supplementary Figure 6 | Distribution of training and test data. Circle represents training data, and asterisk represents test data. Blue and green represents SPT phase, while orange and red represents trivial phase.

By repeating the above process 10 times, we obtained the success probability of phase classification to be 0.945 on average.

Supplementary Note 5: Experimental details

In this Note, we will discuss the details of the experiments and learning processes carried out in the main text.

Predicting properties of the ground state

5.1. Learning the ground state properties of 1D nearest-neighbor random hopping system

In our study, we tried to learn the ground state properties of a 1D nearest-neighbor (NN) random hopping system $H_{\text{hop}}(x) = \sum_i x_i (a_i^\dagger a_{i+1} + a_{i+1}^\dagger a_i)$ where x_i was sampled from [0, 2] and, in this parameter regimes, there are multiple quantum phases. In the theoretical proof regarding the performance of ML model[10], they assumed that both training and test data are within the same phase. However, we expanded the learning task to a more realistic and practical scenario (where it's unknown which phases the given system have) and observed that widely used ML models work effectively even this case with favorable scaling $N_{\text{data}} \sim \mathcal{O}(1/\text{RMSE})$ where RMSE is Root-Mean-Square error of the ML model. Our research demonstrated that the results predicted by theoretical papers are valid not only for numerically generated data but also for data from current noisy quantum computers with error mitigation.

5.1.1 Kernel regression

In the Fig. 2 of main text, we calculated the value of $f(x_{\text{new}})$ for a new point x_{new} using the following formula

$$f(x_{\text{new}}) = \sum_{i=1}^{N_{\text{data}}} \sum_{j=1}^{N_{\text{data}}} k(x_{\text{new}}, x_i)(K + \lambda I)^{-1}_{ij} f(x_j) \tag{17}$$

where λ is hyperparameter, $K_{ij} = k(x_i, x_j')$ is a kernel matrix and I is $N_{\text{data}} \times N_{\text{data}}$ identity matrix.

Using the kernel trick $k(x,x') = \varphi(x)^T \varphi(x)$, and w^* which minimizes the L_2 loss

$$w^* = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{N_{data}} \frac{1}{2} \left| w^T \phi(x_i) - f(x_i) \right|^2 + \frac{\lambda}{2} \|w\|_2^2 \tag{18}$$

where the second term here is used for regularization, the predicted value for x_{new} can be written as $f_{\text{ML}}(x_{\text{new}}) = w^{*T} \varphi(x_{\text{new}})$, and by simplifying this, (Supplementary Eq. (17)) can be derived[19].

In the experiments, ML was conducted using not only the (modified) Dirichlet kernel but also Gaussian kernel and neural tangent kernel (NTK)[28]. The calculation of the NTK function was performed using the software package neural-tangents[29]. The specific expressions for each kernel function are as follows

$\tilde{k}(x,x') = \sum_{i \neq j} \sum_{k_i=-3}^{3} \sum_{k_j=-3}^{3} \cos(\pi(k_i(x_i - x_i') + k_j(x_j - x_j')))$ (modified) Dirichlet kernel (19a)

$\tilde{k}(x,x') = \exp(-\alpha \|x - x'\|_2^2)$ where $\alpha = N_{\text{data}}^2 / \sum_{i=1}^{N_{\text{data}}} \sum_{j=1}^{N_{\text{data}}} \|x_i - x_j\|_2^2$ Gaussian kernel (19b)

$\tilde{k}(x,x') = k^{(\text{NTK})}(x,x')$ NTK (19c).

We normalized $\tilde{k}(x,x')$ into $k(x,x') = \tilde{k}(x,x') / \sqrt{\tilde{k}(x,x)\tilde{k}(x',x')}$ and used $k(x,x')$ as a kernel function in our ML procedures.

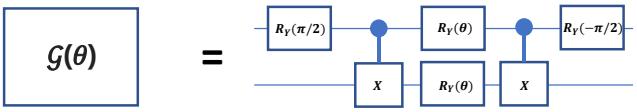
In our experimental results, the commonly used Gaussian kernel outperformed the Dirichlet kernel, which was used in proof[10]. A qualitative interpretation may be considered. The Gaussian kernel assigns larger weights only when the input parameters x and x' are close in L_2 distance. Conversely, as observed in many phase diagrams, parameters x from different phases are usually far apart, except at the phase transition point. This spatial separation causes the importance of data points in different phases to decrease exponentially with the square of the distance, due to the Gaussian kernel. Therefore, even in training data that includes different phases, the prediction for a new point x_{new} often relies on the training data in the vicinity of x_{new}, which approximately satisfies the assumption in[10].

5.1.2 Used device and qubits for the experiment

For the Fig. 2 of main text, we utilized the *ibm_sherbrooke* device and used the 12 qubits corresponding to [87, 88, 89, 74, 70, 69, 68, 67, 66, 73, 85, 84].

5.1.3 Gate decompositions for Givens rotation.

In the experiment, a total of 36 Givens rotation blocks were needed, and each block required 2 CNOT gates (Supplementary Figure 7). As mentioned, for implementing on the IBM quantum hardware, we decompose CNOT and R_Y into $\{R_Z(\theta), SX, X, ECR(\pi/2)\}$

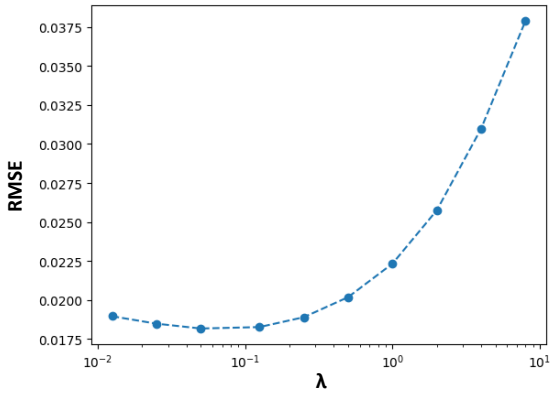


Supplementary Figure 7 | Givens rotation block ($\mathcal{G}(\theta) = \mathcal{U}(G(\theta))$) decompositions. We decomposed the Givens rotation block using two CNOTs. In the experiment, we implemented each CNOT by decomposing it into one ECR(π/2) and single qubit gates.

5.1.4 The process of selecting hyperparameter (λ) and the ML model performance from different hyperparameters

In the ML, we used a total of 200 data points for training ML model and 10,000 test data points for evaluating the model's performance. When determining hyperparameters λ in equation (2) of the main text, we used 1,000 validation data points that did not overlap with the test data. Among [0.0125, 0.025, 0.05, 0.125, 0.25, 0.5, 1, 2, 4, 8], we chose the λ that produced the smallest Root-Mean-Square Error (RMSE) on the validation data. We also trained ML model with non-optimal hyperparameter λ and observed that training generally worked well across a wide range of λ (Supplementary Figure 8). We used the hyperparameter λ = 0.05 in

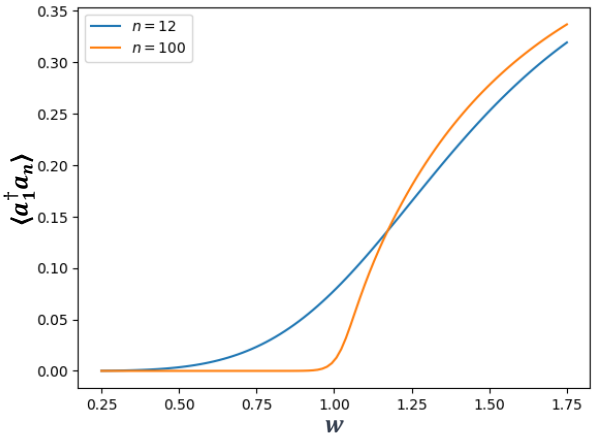
the Fig. 2 of the main text.



Supplementary Figure 8 | Monitoring ML performance as we varied the hyperparameter (λ).

5.1.5 Edge correlation of Su–Schrieffer–Heeger (SSH) system.

The SSH Hamiltonian $H_{\mathrm{SSH}} = \sum_i (v a_{2i-1}^\dagger a_{2i} + w a_{2i}^\dagger a_{2i+1}) + h.c.$ exhibits either trivial or topological phases depending on the relative magnitudes of v and w in the Hamiltonian. Among the order parameters to distinguish these phases, edge correlation $a_1^\dagger a_n$ was employed in the Fig. 2g. By fixing $v = 1$ and varying w between 0.25 and 1.75, we calculated the value of $\langle a_1^\dagger a_n \rangle$ with $n =$ 100 sites and $n = 12$ sites (used in the experiment of the main text) by classical simulation. As shown in the Supplementary Figure 9, there is a sharp change near the $w = 1$ which is consistent with known phase boundary between trivial and topological phase (trivial when $v < w$ and topological when $v > w$).

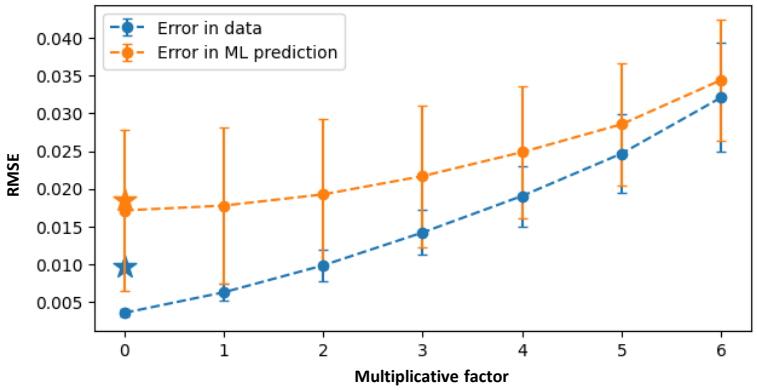


Supplementary Figure 9 | Edge correlation $\langle a_1^\dagger a_n \rangle$.

5.1.6. Noisy simulation results

We conducted noisy simulations for the experiments in the main text. We investigated how the performance of the ML model changed as gate and measurement errors increase. In noisy simulation, in addition to the existing hardware errors, we added depolarizing error channels with error rate of p_{single} and p_{two} to single-qubit and two-qubit gates and measurement error with error rate p_{m}. Through this simulation, we can confirm that using various error mitigation

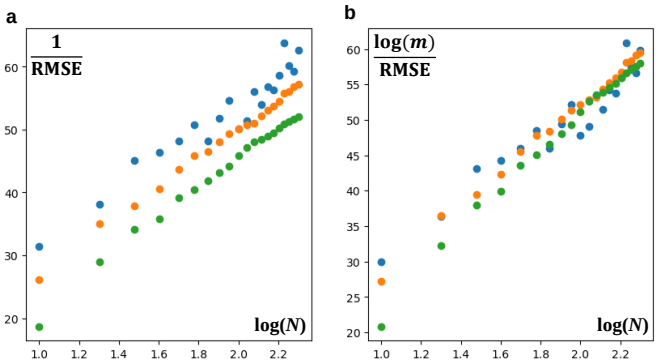
techniques introduced in Supplementary Note 3 leads to better ML model performance (Supplementary Figure 10).



Supplementary Figure 10 | Noisy simulation. The blue dots represent the RMSE for training data, while the orange dots indicate the RMSE for the 10,000 test data points obtained from the trained ML model. The asterisks represent the RMSE in the training data and ML prediction results used in the main text. The error bars represent the standard deviation of 100 instances. In the noisy simulation, we imposed (p_{single}, p_{two}, p_{measure}) = (0.001, 0.01, 0.01) × (multiplicative factor).

5.1.7. Comparison of the prediction error for different m

We have examined the scaling between N and prediction error in the main text while fixing the number of model parameters m.



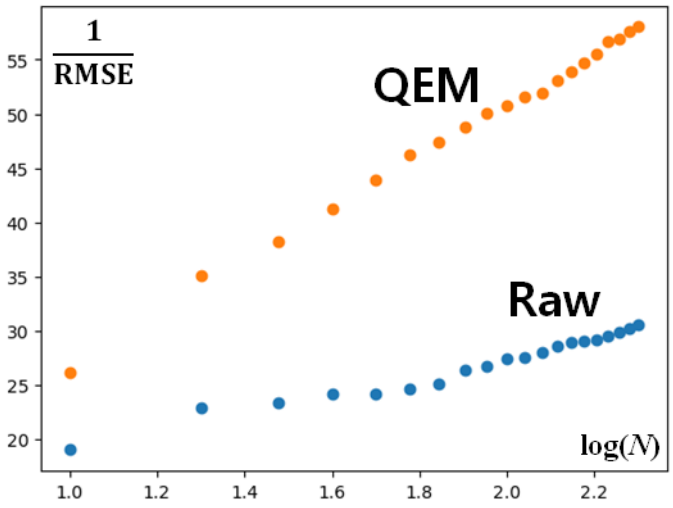
Supplementary Figure 11| Comparison of the ML model's prediction error for different values of m. Blue, orange and green dots correspond m = 9, m = 11, m = 13 respectively

As shown in the Supplementary Figure 11b, we observe a reasonable agreement in the scaling

of log(m)/RMSE ~ log(N) across different values of m, indicating that classical ML algorithms function effectively even as the system's number of parameters, m, increases in the presence of errors in the data (Ideally, the trends for different m should overlap, but there were slight deviations due to differences in errors of training data of each case of m).

5.1.8. Comparison of the prediction error between with and without error mitigations

We have conducted an analysis comparing the outcomes of training with raw data versus error mitigated data and have observed that through error mitigation, not only can we obtain training data with lower errors, but as seen in the Supplementary Figure 12 (m = 11), we can also achieve more favorable scaling of (N, error).



Supplementary Figure 12| Comparison of the ML model's prediction error with and without error mitigations.

Classifying quantum phase

In the main text, we first conducted experiments to distinguish between SPT and trivial phases, and then, topologically ordered and trivial phases. Here, we will describe the details of the experiments and ML processes. For both experiments, we used kernel principal component analysis (PCA) and support vector machine (SVM) as classical ML algorithms. These two methods have been widely used in data analysis[19]. We first performed kernel PCA using the shadow kernel[10] developed for learning non-linear quantities of the quantum state ρ. After performing kernel PCA, each data point can be represented using principal eigenvectors as the bases. We focused up to the second principal components, reducing the input data dimension to 2. In the 2D space, we executed kernel SVM with the Gaussian kernel, utilizing the software package sklearn[30]. Through ML, we can obtain the phase boundary in the 2D projected space, and this enables us to predict the phase of a new quantum state.

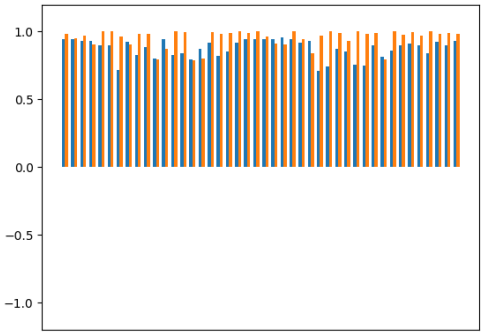
5.2. Distinguishing a short-range entangled state from a trivial one

5.2.1. Used device and qubits for the experiment

For the images in Fig. 3 of the main text, we utilized the *ibm_sherbrooke* device and used the 44 qubits corresponding to [126, 112, 108, 107, 106, 105, 104, 103, 102, 101, 100, 99, 98, 91, 79, 80, 81, 72, 62, 61, 60, 59, 58, 71, 77, 76, 75, 90, 94, 95, 96, 109, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125].

5.2.2. Expectation values of stabilizers of the resulting cluster state.

We used the 1D cluster state corresponding to the ground state of $H_{ZXZ} = -\sum_i Z_{i-1}X_iZ_{i+1}$ with periodic boundary condition as the fixed points of the SPT phase. To verify how well the state was prepared on a quantum computer, we conducted an experiment measuring stabilizers $\langle Z_{i-1}X_iZ_{i+1}\rangle$ shown in the Supplementary Figure 13.



Supplementary Figure 13 | Measurement result of the stabilizers for cluster state. The blue bar represents the results obtained from raw data, and the orange bar corresponds to the measurement-error-mitigated results. Through measurement error mitigation, we can compensate for the errors occurred during the measurement, as the average value shifts from 0.876 to 0.953.

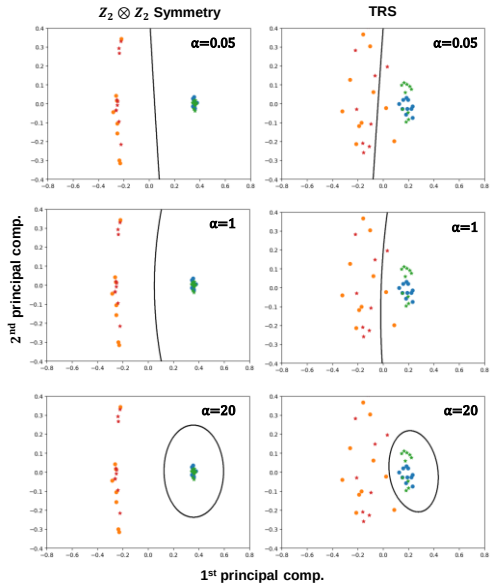
5.2.3. Random unitary for generating data

In the experiment of main text, we used a total of 20 data points for training ML model, with 10 points each from the SPT and trivial phases. We employed the SPT phase protected by $\mathbb{Z}_2\otimes\mathbb{Z}_2$ symmetry generated by $X_{\text{even(odd)}} = \prod_{i=\text{even(odd)}}X_i$ or time reversal symmetry (TRS) generated by $\mathcal{T} = (\prod_i X_i)K$ where X_i, Z_i are Pauli operators at the site i and K denotes complex-conjugation. We created different states within the same phase by applying single or two qubit gates sampled from certain set that respects the given symmetry[9]. In the case of $\mathbb{Z}_2\otimes\mathbb{Z}_2$ symmetry, we uniformly sampled $P\in\{I, X_1, X_2, X_1X_2\}$ and $\theta\in[0, 2\pi]$, and then applied $e^{i\theta P}$ to the corresponding qubits. For the TRS case, we followed the same approach, but sampling P

$\in \{I, Z_1, Z_2, Z_1Y_2, Z_2Y_1, Z_1X_2, X_1Z_2\}$. We applied two layers of random unitary in each symmetry case. In particular, for the TRS case, it has been proven that when random unitary is sampled from the aforementioned set, it is hard to distinguish between the SPT and trivial phases using quantities in the form of $\mathrm{Tr}(O\rho)$[9]. Moreover, in actual experiments, the SOP value exponentially decreases as the length of SOP increases due to the measurement and gate errors, making distribution of SOP even denser near 0.

5.2.4. Hyperparameter in ML.

The shadow kernel in (SE7) has two hyperparameters, τ and γ, and in the main experiment, we used $\tau = \gamma = 1$. After kernel PCA, kernel SVM with a Gaussian kernel was employed with hyperparameter $\alpha = 1/(n_{\text{feature}}\sum_{ij}|X_{ij}\text{-E(X)}|^2)$, where $n_{\text{feature}} = 2$ in our case, $\text{E(X)} = 1/(n_{\text{feature}}\cdot N_{\text{data}})\sum_{ij}X_{ij}$, and X_{ij} is the j-th feature value of the i-th data. Additionally, we investigated how the the phase boundary changed as we varied the hyperparameter α in the Supplementary Figure 14.

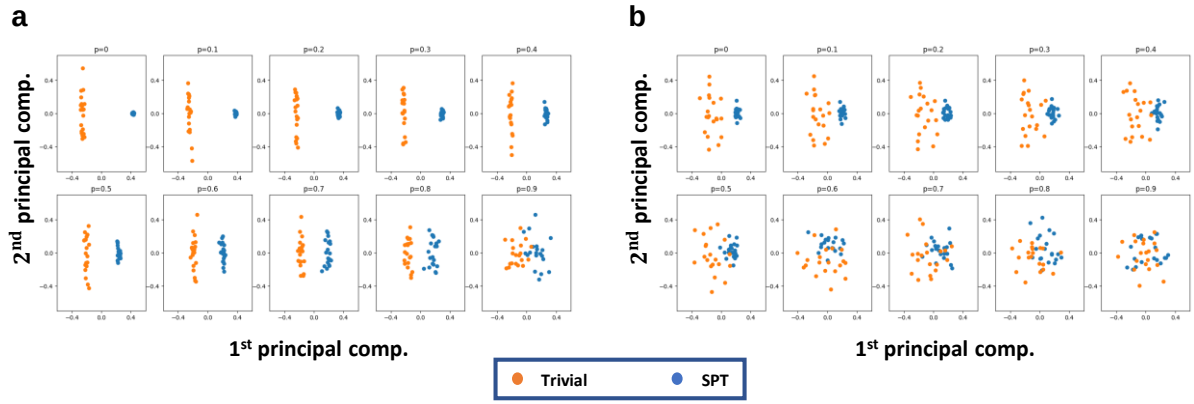


Supplementary Figure 14 | Change of phase boundary depending on the hyperparameter α of the Gaussian kernel.

For Fig. 3e and 3f of the main text, the hyperparameter of Gaussian kernel is $\alpha = 7.87$ in the case of $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ symmetry and $\alpha = 18.8$ in the TRS case.

5.2.5. Noisy simulation

We conducted noisy simulations for the experiments in the main text. To carry out noisy simulations, we employed the depolarizing error channel $D_p(\rho) = (1\text{-}p)\rho + pI_d/d$ where $d = 2^n$ and I_d is $d \times d$ identity matrix. We investigated how the the distribution of the data changed as we varied depolarizing error rate p (Supplementary Figure 15).

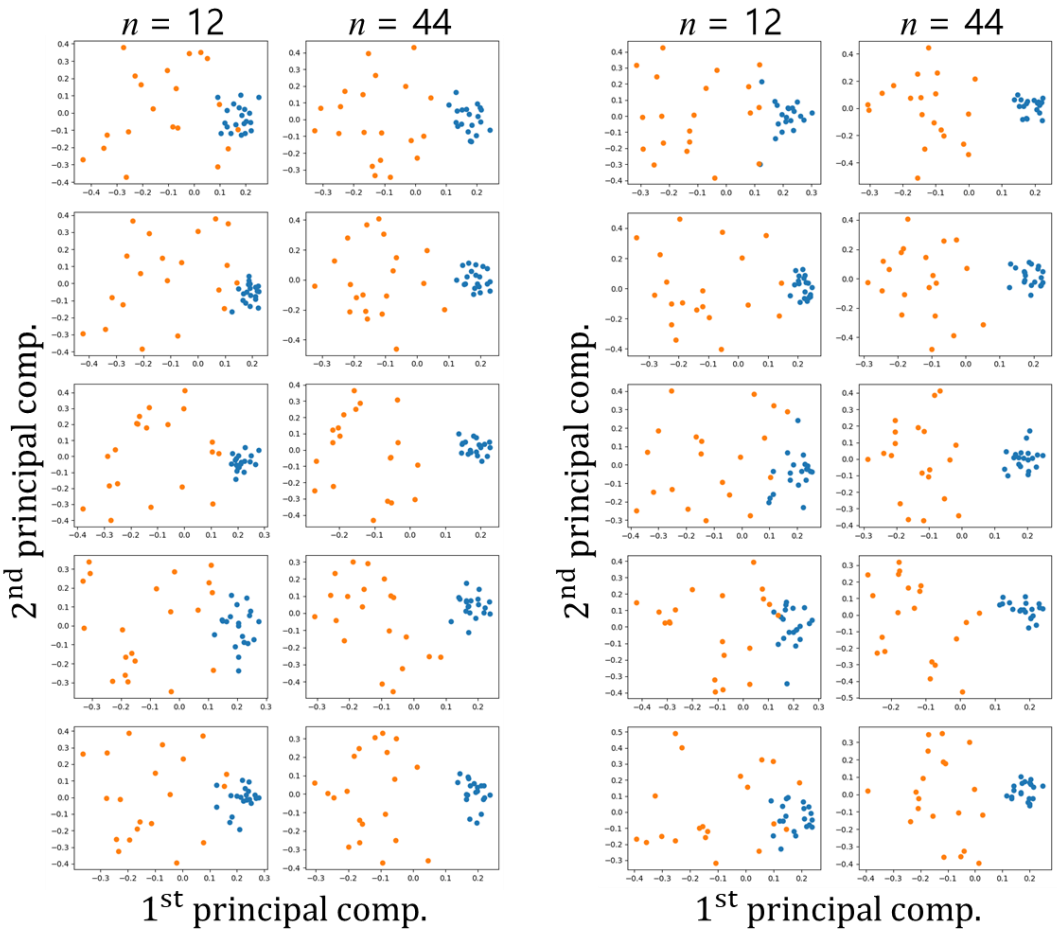


Supplementary Figure 15 | **a**, Change of data distribution for $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ symmetry case depending on the depolarizing error rate p. **b**, Change of data distribution for time reversal symmetry (TRS) case depending on the depolarizing error rate p.

5.2.6. Comparison of data distribution for different system size n

The dependency of the algorithm's computational time on n appears when calculating the shadow kernel $k^{(\text{shadow})}(S_T(\rho_1), S_T(\rho_2))$ and is proportional to $\mathcal{O}(n)$. Therefore, even as the system size increases, the computational time only increases linearly. In addition, our further experiments compared the distribution of experimental data for SPT and trivial phases across different system sizes n, we found that when the number of layers of symmetric random unitary

(d_{LU}) and T are the same, the larger the system size, the more clearly the quantum phase separates.



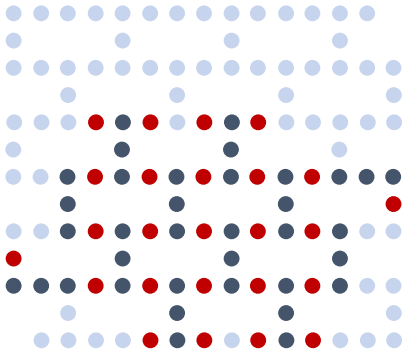
Supplementary Figure 16 | The distribution of experimental data for SPT with time reversal symmetry and trivial phase when $T = 100$, for $n = 12$ and $n = 44$. The x-axis corresponds to the first principal component, and the y-axis corresponds to the second principal component. Orange dots correspond to the trivial phase and blue dots to the SPT phase.

In Supplementary Figure 16, the phase boundary is relatively less clear for $n = 12$ compared to $n = 44$. In other words, as the system size increases, the task of separating quantum phases can be accomplished with a smaller T, and considering that the computational time of the algorithm is proportional to $\mathcal{O}(T^2)$, cases with a smaller n and larger T might actually take more computational time.

5.3. Distinguishing a topological phase from trivial phase

5.3.1. Used device and qubits for the experiment

In the experiment for distinguishing between topologically ordered and trivial phases, we used *ibm_sherbrooke* and employed 61 qubits corresponding to [77, 76, 101, 102, 103, 105, 106, 107, 78, 92, 93, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 72, 73, 74, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 53, 54, 55, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 34, 35, 36, 23, 24, 25, 29, 28, 27, 31, 32]. Among them, 25 qubits were utilized to store information for the $|0_L\rangle$ with a code distance of 5, while the remaining 36 qubits were used as ancilla qubits (Supplementary Figure 17).

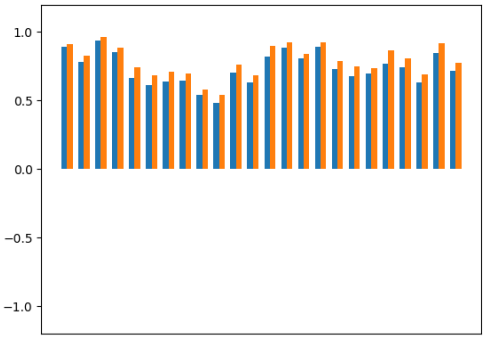


Supplementary Figure 17 | Used 61 qubits for the experiment out of 127 qubits in *ibm_sherbrooke*. The 25 red colored qubits were used to store information for the $|0_L\rangle$ state with $d_{code} = 5$ while the remaining 36 qubits served as ancilla qubits needed for measurement-assisted state preparation method.

5.3.2. Stabilizer measurement result

We used the $|0_L\rangle$ state in equation (4) of the main text corresponding to one of the ground states of $H = -\sum_s A_s - \sum_p B_p$, $(A_s = \prod_{i \in s} Z_i,\ B_p = \prod_{j \in p} X_j)$ where s is a Z-plaquette and p is a X-plaquette as the fixed-point state of the topologically ordered phase. To verify how well the state was prepared on a quantum computer, we conducted an experiment measuring stabilizers (Supplementary

Figure 18).



Supplementary Figure 18 | Measurement result of the stabilizers for $|0_L\rangle$ state. The blue bars represent the results obtained from raw data, and the orange bars correspond to the measurement-error-mitigated results. Through measurement error mitigation, we can compensate for the errors occurred during the measurement, as the average value shifts from 0.733 to 0.784.

5.3.3. Random unitary for generating data

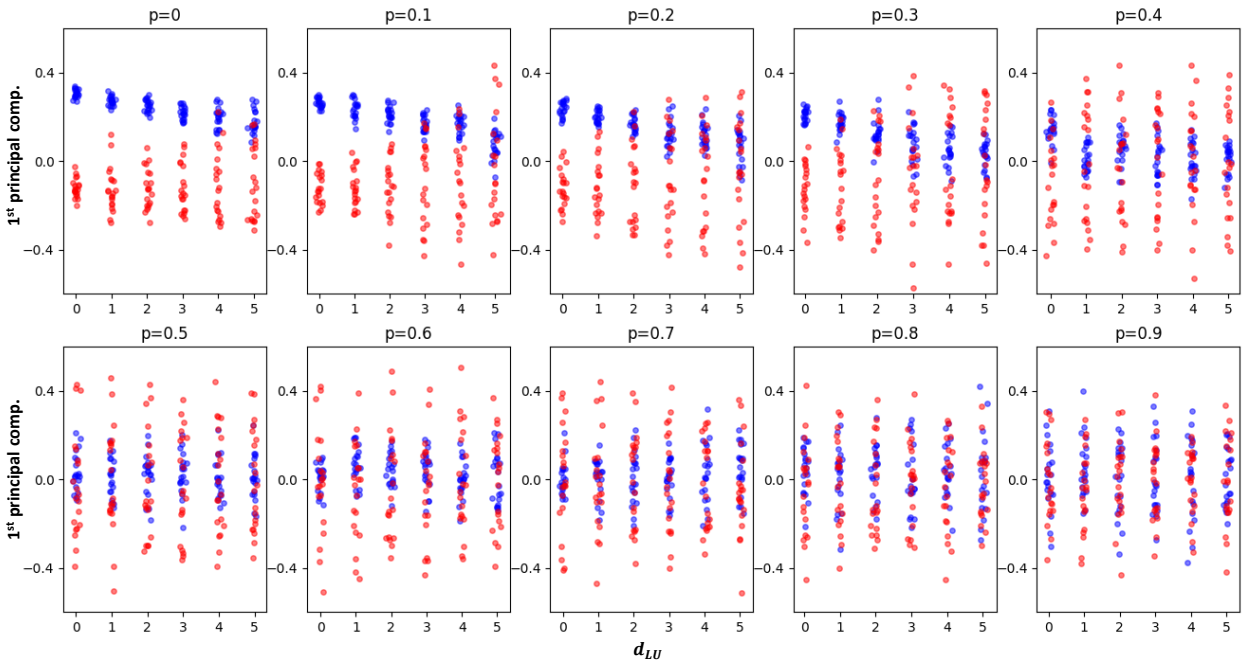
In the experiment of the main text, we applied local random unitary to create 10 different training data in each phase. In the topologically ordered phase, we applied the tensor product of single qubit gates due to hardware qubit connectivity, while in the trivial phase, we applied local random unitary up to 5 layers where each layer consists of CX gates and random single qubit gates. Although we only applied relatively simple tensor product of single qubit gates for generating data in the topologically ordered phase, it is known that it is impossible to distinguish a topologically ordered phase from a trivial phase using (global) linear observable corresponding to $\mathrm{Tr}(O\rho)$[10]. Therefore, using conventional topological string order parameters is highly likely to fail in cases like ours, where the target state deviates from the fixed-point state.

5.3.4. Hyperparameter in ML.

The shadow kernel in (SE7) has two hyperparameters, τ and γ, and in the experiment of main text, we used $\tau = \gamma = 1$.

5.3.5 Noisy simulation

We conducted noisy simulations for the experiments obtained from the quantum computer in the main text. To carry out noisy simulations, we employed the depolarizing error channel $D_p(\rho) = (1\text{-}p)\rho + pI_n/d$ where $d = 2^n$ and I_n is $d \times d$ identity matrix. We investigated how the distribution of the data changed as we varied depolarizing error rate p (Supplementary Figure 19).



Supplementary Figure 19 | Change of data distribution depending on the depolarizing error rate p.

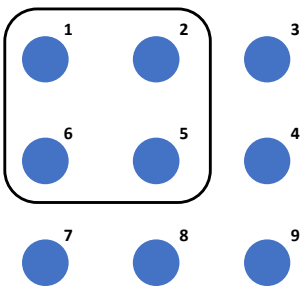
5.4. Unlocking the black box of the ML model

Although our analysis is based on methodologies with mathematically guaranteed performance, there are instances in actual experiments or simulations where certain conditions

deviate slightly from the assumptions used in mathematical proofs[10]. For instance, in the problem of predicting the properties of the ground state, the data used for training must all be in the same phase, and the system we want to predict through the ML model must also be in the same phase as training data to guarantee the proven mathematical performance. However, our ML task used data from a 1D nearest neighbor random hopping system containing multiple quantum phases, not strictly satisfying the required mathematical assumptions. Additionally, in the classification task, we used a modified shadow kernel (SE7) for numerical stability, which deviates slightly from the original one (equation (3)) in the main text. In such cases, we might question the performance of the ML model and may require information about what aspects of the data ML model has considered during the ML procedure. In this context, we conducted an experiment to extract what structure the ML model employed within the data in classifying between topologically ordered and trivial phases in a 9-qubit system.

5.4.1. Used device and qubits for the experiment

For the Fig. 5 of main text, we utilized the *ibm_sherbrooke* device and used the 9 qubits corresponding to [117, 118, 119, 120, 121, 122, 123, 124, 125] (Supplementary Figure 20).



Supplementary Figure 20 | Out of 9 qubits, the [1, 2, 5, 6] sub-system was used for ML

5.4.2. Random unitary for generating data

In the experiment, to create states in the topologically ordered and trivial phases, we applied a random unitary consisting of two layers to the fixed-point states of each phase. Each layer is composed of CX gates and arbitrary single qubit gates.

5.4.3. Feature mapping

To extract phase classifier from the ML model, we used the following feature vector $\varphi(\rho) \in \mathrm{R}^{15}$, which maps the given state to a relatively lower-dimensional space.

$$\begin{aligned}\varphi(\rho) = [&S^{(2)}(\rho_{[1]}), S^{(2)}(\rho_{[2]}), S^{(2)}(\rho_{[5]}), S^{(2)}(\rho_{[6]}), S^{(2)}(\rho_{[1,2]}),\\ &S^{(2)}(\rho_{[1,5]}), S^{(2)}(\rho_{[1,6]}), S^{(2)}(\rho_{[2,5]}), S^{(2)}(\rho_{[2,6]}), S^{(2)}(\rho_{[5,6]}),\\ &S^{(2)}(\rho_{[1,2,5]}), S^{(2)}(\rho_{[1,2,6]}), S^{(2)}(\rho_{[1,5,6]}), S^{(2)}(\rho_{[2,5,6]}), S^{(2)}(\rho_{[1,2,5,6]})]^{\mathrm{T}}\end{aligned} \tag{20}$$

where ρ_A (= $\mathrm{Tr}_{\neg A}(\rho)$) is the reduced density matrix (RDM) obtained by tracing out all qubit indices except those corresponding to A and $S^{(2)}(\rho) = -\log_2(\mathrm{Tr}(\rho^2))$ is a Renyi-2 entanglement entropy. In the experiment described in the main text, due to the relatively small number of qubits, the Maximum-likelihood estimator quantum state tomography (MLE-QST)[31] was used to obtain each RDM in the feature vector $\varphi(\rho)$. Unfortunately, when calculating $\mathrm{Tr}(\rho_A^2)$ using this method, exponential resources in samples and computational time are necessary as the size of A increases. To circumvent this issue, the swap test[33] can be utilized. Recently, a method[32] combining randomized measurement[7] and the swap test has been introduced. Implementing this to calculate the purity in larger regions of the A leads to promising subsequent research.

5.4.4. Improvement in ML performance with measurement error mitigation (MEM).

We employed the MEM introduced earlier to offset the errors that occurred during the data acquisition for MLE-QST. By comparing the ML model's performance on test data after training with raw data and measurement-error-mitigated data, we confirmed that the successful phase classification rate increased from 0.942 to 0.959 when MEM was applied.

5.4.5. Comparison between the classifier obtained from ML and the Renyi-2 TEE.

To compare the performance of the classifier obtained from ML, $f_{\mathrm{ML}}(\rho)=w_{\mathrm{ML}}^{\mathrm{T}}\varphi(\rho)+w_{0,\mathrm{ML}}$ (w_{ML} = [0.0780, 0.0736, 0.0232, 0.0342, 0.230, 0.103, 0.113, 0.0786, 0.0977, 0.091, 0.235, 0.254, 0.172, 0.0152, 0.171]$^{\mathrm{T}}$, $w_{0,\mathrm{ML}}$ = -2.23), and the existing $f_{\mathrm{TEE}}(\rho)=w_{\mathrm{TEE}}^{\mathrm{T}}\varphi(\rho)+w_{0,\mathrm{TEE}}$ ($w_{\mathrm{TEE}}=[1,0,0,1,0,0,-1,1,0,0,-1,0,0,-1,1]^{\mathrm{T}}$, $w_{0,\mathrm{TEE}}=0.1$ which is set to minimize classification error), We generated 6000—3000 from each phase—test data via classical simulation. An error $\varepsilon\in$[-0.1, 0.1] was added to each $S^{(2)}(\rho_A)$ of $\varphi(\rho)$ in (Supplementary Eq. (20)) to emulate real-world experimental conditions.

Supplementary References

1. Jurcevic, P. et al. Demonstration of quantum volume 64 on a superconducting quantum computing system. Quantum Sci. Technol. 6, 025020 (2021).
2. Nation, P. D. & Treinish, M. Suppressing Quantum Circuit Errors Due to System Variability. PRX Quantum 4, 010327 (2023).
3. Gadi, A. et al. Qiskit: An open-source framework for quantum computing. (2019).
4. Haah, J., Harrow, A. W., Ji, Z., Wu, X. & Yu, N. Sample-optimal tomography of quantum

states. IEEE Trans. Inform. Theory 1–1 (2017) doi:10.1109/TIT.2017.2719044.

5. Nielsen, M. A. & Chuang, I. L. Quantum computation and quantum information. (Cambridge University Press, 2010).

6. Resch, K. J., Walther, P. & Zeilinger, A. Full Characterization of a Three-Photon Greenberger-Horne-Zeilinger State Using Quantum State Tomography. Phys. Rev. Lett. 94, 070402 (2005).

7. Huang, H.-Y., Kueng, R. & Preskill, J. Predicting many properties of a quantum system from very few measurements. Nat. Phys. 16, 1050–1057 (2020).

8. Zhang, T. et al. Experimental Quantum State Measurement with Classical Shadows. Phys. Rev. Lett. 127, 200501 (2021).

9. Liu, Y.-J., Smith, A., Knap, M. & Pollmann, F. Model-Independent Learning of Quantum Phases of Matter with Quantum Convolutional Neural Networks. Phys. Rev. Lett. 130, 220603 (2023).

10. Huang, H.-Y., Kueng, R., Torlai, G., Albert, V. V. & Preskill, J. Provably efficient machine learning for quantum many-body problems. Science 377, eabk3333 (2022).

11. Kim, Y. et al. Scalable error mitigation for noisy quantum circuits produces competitive expectation values. Nat. Phys. 19, 752–759 (2023).

12. Sung, K. J., Rančić, M. J., Lanes, O. T. & Bronn, N. T. Simulating Majorana zero modes on a noisy quantum processor. Quantum Sci. Technol. 8, 025010 (2023).

13. Wallman, J. J. & Emerson, J. NoiSEtailoring for scalable quantum computation via randomized compiling. Phys. Rev. A 94, 052325 (2016).

14. Google AI Quantum and Collaborators et al. Hartree-Fock on a superconducting qubit quantum computer. Science 369, 1084–1089 (2020).

15. Jiang, Z., Sung, K. J., Kechedzhi, K., Smelyanskiy, V. N. & Boixo, S. Quantum Algorithms to Simulate Many-Body Physics of Correlated Fermions. Phys. Rev. Applied 9, 044036

(2018).
16. Satzinger, K. J. et al. Realizing topologically ordered states on a quantum processor. Science 374, 1237–1241 (2021).
17. Lu, T.-C., Lessa, L. A., Kim, I. H. & Hsieh, T. H. Measurement as a Shortcut to Long-Range Entangled Quantum Matter. PRX Quantum 3, 040337 (2022).
18. Cong, I., Choi, S. & Lukin, M. D. Quantum convolutional neural networks. Nat. Phys. 15, 1273–1278 (2019).
19. Bishop, C. M. Pattern recognition and machine learning. (Springer, 2006).
20. Pesah, A. et al. Absence of Barren Plateaus in Quantum Convolutional Neural Networks. Phys. Rev. X 11, 041011 (2021).
21. Herrmann, J. et al. Realizing quantum convolutional neural networks on a superconducting quantum processor to recognize quantum phases. Nat. Commun. 13, 4144 (2022).
22. Vidal, G. A class of quantum many-body states that can be efficiently simulated. Phys. Rev. Lett. 101, 110501 (2008).
23. Evenbly, G. & Vidal, G. Tensor Network Renormalization. Phys. Rev. Lett. 115, 180405 (2015).
24. McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R. & Neven, H. Barren plateaus in quantum neural network training landscapes. Nat. Commun. 9, 4812 (2018).
25. Pollmann, F. & Turner, A. M. Detection of symmetry-protected topological phases in one dimension. Phys. Rev. B 86, 125441 (2012).
26. Elben, A. et al. Many-body topological invariants from randomized measurements. Sci. Adv. 6, eaaz3666 (2020).
27. Elben, A. et al. Mixed-State Entanglement from Local Randomized Measurements. Phys. Rev. Lett. 125, 200501 (2020).
28. Jacot, A., Gabriel, F. & Hongler, C. Neural Tangent Kernel: Convergence and

Generalization in Neural Networks. Preprint at http://arxiv.org/abs/1806.07572 (2020).

29. Novak, R. et al. Neural Tangents: Fast and Easy Infinite Neural Networks in Python. Preprint at http://arxiv.org/abs/1912.02803 (2019).

30. Pedregosa, F. et al. Scikit-learn: Machine Learning in Python. MACHINE LEARNING IN PYTHON.

31. Smolin, J. A., Gambetta, J. M. & Smith, G. Efficient Method for Computing the Maximum-Likelihood Quantum State from Measurements with Additive Gaussian Noise. Phys. Rev. Lett. 108, 070502 (2012).

32. Zhou, Y. & Liu, Z. A hybrid framework for estimating nonlinear functions of quantum states. Preprint at http://arxiv.org/abs/2208.08416 (2023).

33. Buhrman, H., Cleve, R., Watrous, J. & De Wolf, R. Quantum Fingerprinting. Phys. Rev. Lett. 87, 167902 (2001).