

Hamiltonian learning via inverse physics-informed neural networks

Jie Liu^{1,2,3} and Xin Wang^{1,2,*}

¹Department of Physics, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong SAR, China

²City University of Hong Kong Shenzhen Research Institute, Shenzhen, Guangdong 518057, China

³Quantum Science Center of Guangdong-Hong Kong-Macao Greater Bay Area, Shenzhen, Guangdong 518045, China



(Received 12 June 2025; accepted 6 October 2025; published 5 November 2025)

Hamiltonian learning (HL), enabling precise estimation of system parameters and underlying dynamics, plays a critical role in characterizing quantum systems. However, conventional HL methods face challenges in noise robustness and resource efficiency, especially under limited measurements. In this work, we present Inverse Physics-Informed Neural Networks for Hamiltonian Learning (iPINN-HL), an approach that incorporates the Schrödinger equation as a soft constraint via a loss function penalty into the ML procedure. This formulation allows the model to integrate both observational data and known physical laws to infer Hamiltonian parameters with greater accuracy and resource efficiency. We benchmark iPINN-HL against a deep-neural-network-based quantum state tomography method [denoted as DNN-HL (Deep Neural Network for Hamiltonian Learning)] and demonstrate its effectiveness across several different scenarios, including one-dimensional spin chains, cross-resonance gate calibration, crosstalk identification, and real-time compensation to parameter drift. Our results show that iPINN-HL can approach the Heisenberg limit and exhibits robustness to noises, while outperforming DNN-HL in accuracy and resource efficiency. Therefore, iPINN-HL is a powerful and flexible framework for quantum system characterization for practical tasks.

DOI: [10.1103/nx97-zjdf](https://doi.org/10.1103/nx97-zjdf)

I. INTRODUCTION

Over the past few decades, advances in quantum technologies, particularly in the field of quantum control, have enabled substantial progress in precise manipulation of quantum states [1–5]. Nonetheless, achieving even higher precision remains crucial for the design of quantum gates, refinement of control strategies, and robust implementation of high-fidelity quantum operations under realistic, noisy conditions. Such improvements are essential for realizing practical quantum error correction and for scaling up viable quantum computing architectures [6,7]. Reaching this goal demands more advanced system characterization techniques capable of accurately identifying device parameters and diagnosing noise sources [8–12].

Current quantum devices are subject to various sources of noise, including decoherence, readout errors, and unintended interactions between neighboring qubits [13–15]. Identifying and quantifying these error sources is a key step toward effective device characterization. Moreover, quantum hardware often requires frequent recalibration to mitigate parameter drift over time [16], which poses significant challenges to the efficiency and stability of system characterization procedures.

System characterization and Hamiltonian learning (HL) are intrinsically related, as both aim to uncover the underlying dynamics of quantum systems [17]. In HL, the objective is to infer the Hamiltonian that governs the evolution of a system from experimental data, using limited classical and quantum resources [18–21]. In quantum mechanics, the evolution is governed by the Schrödinger equation, in which both the Hamiltonian and various noise terms determine the dynamics of the system. Consequently, quantum system characterization can be formulated as an HL problem, where the goal is to extract an accurate description of the dynamics of the system from data. Learning the Hamiltonian provides access to system parameters, interactions, and noise sources, thereby laying the groundwork for effective calibration and precise control.

However, it is unavoidable that the resources required to learn a generic many-body Hamiltonian grow exponentially with the system size [22]. When the system size is fixed, the estimation accuracy scales polynomially with the resources as $m^{-\ell}$ and is bounded by the standard quantum limit with $\ell = 1$ or by the Heisenberg limit with $\ell = 2$, where m quantifies the amount of resources used in estimation. While it is possible for an optimized HL protocol to surpass the standard quantum limit, practical limitations such as imperfect state preparation, manipulation, and various sources of noise in real devices often hinder the full attainment of the Heisenberg limit. Despite these challenges, it remains advantageous to minimize the resources needed to achieve a given level of accuracy within the HL framework. This focus on resource efficiency is a central aim of this paper.

Machine learning (ML) has been widely applied in HL. In HL, the requirement to gather extensive measurement data

*Contact author: x.wang@cityu.edu.hk

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

over time to accurately characterize system dynamics provides an ideal context for ML. As a data-driven approach, ML is particularly effective at identifying patterns and inferring hidden parameters from large datasets, even when faced with noise or incomplete information. Recent studies have applied various ML techniques to extract system Hamiltonians both numerically and experimentally, including recurrent neural networks [23], autoencoders [24], Bayesian inference [25], and active learning [26]. For example, neural networks can be trained in real time to predict the quantum trajectory of a superconducting qubit undergoing unitary evolution [23]. While traditional data-driven ML methods have demonstrated considerable success in HL protocols, they often lack direct integration of the physical principles that govern the system. This limitation significantly increases the data demand for achieving reliable generalization and hinders the adaptability of the model when extrapolating beyond the observed data regime. These issues are particularly pronounced in HL, where data collection can be costly, and densely sampling over the entire time period is impractical.

Recently, Physics-Informed Neural Networks (PINNs), an interesting approach in ML that goes beyond purely data-driven methods, address this gap by incorporating the underlying physical equations into the learning process [27–31]. By integrating *a priori* physical knowledge—such as the Schrödinger equation—into ML, PINNs accelerate convergence, enhance accuracy, and enable more reliable extrapolation beyond observed data [32,33]. Reference [34] applies PINN to infer material properties in heat conduction and to reconstruct flow fields in fluid dynamics. PINN is also used in Ref. [35] to reconstruct permittivity and shape of nanostructures from observed scattering data. This approach is particularly advantageous in scenarios where data are sparse or noisy, as the embedded physical principles help regularize the model and guide it toward physically meaningful solutions.

To utilize PINN in HL, we propose an algorithm called inverse Physics-Informed Neural Network for HL (iPINN-HL). In iPINN-HL, the problem of HL is first reformulated as the inverse problem of Partial or Ordinary Differential Equation (PDE/ODE). Unlike the forward problem of PDE/ODEs, where we solve the equations with known parameters and boundary conditions to numerically find the solution at various time points, the goal of the inverse problem is to estimate the unknown parameters in the PDE/ODE such that the numerical solution obtained under these estimated parameters aligns with experimental observations. HL can be naturally formulated as an inverse problem, governed by the Schrödinger equation, with the Hamiltonian and noise terms treated as unknown parameters.

In this work, we numerically evaluate the performance of iPINN-HL across a range of tasks. We find that iPINN-HL can approach the Heisenberg limit under ideal conditions, i.e., with unconstrained state preparation and measurement and purely unitary dynamics. We further evaluate its performance in practical calibration tasks, including cross-resonance (CR) gate calibration, crosstalk quantification, and parameter-drift compensation. Our results show that it consistently outperforms the purely data-driven baseline, Deep Neural Network for Hamiltonian Learning (DNN-HL), which estimates the unknown parameters using a deep neural network (DNN).

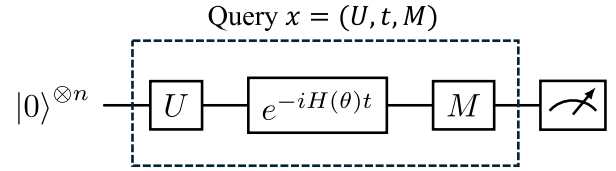


FIG. 1. Circuit illustration of the quantum query model. After the query $x = (U, t, M)$ is input to the system, the output is an n -bit string.

Our iPINN-HL framework introduces soft physics-informed constraints through the loss function, providing flexibility in handling noisy and sparse quantum data. In comparison, structure-preserving networks such as SympNets [36] enforce Hamiltonian dynamics exactly by preserving symplectic structure, ensuring energy conservation but at the cost of architectural complexity. Hamiltonian Neural Networks [37] offer another alternative by learning dynamics via time-derivative prediction, balancing generality with accuracy. Unlike these strict formulations, iPINN-HL favors robustness and adaptability over exact Hamiltonian preservation, making it particularly suitable for quantum systems where data acquisition is costly, as demonstrated in Sec. III.

The paper is organized as follows: In Sec. II A, we provide an overview of HL, introducing the theoretical framework of DNN-HL and the concept of inverse problems. We then discuss the principles of PINNs and their relevance to HL tasks. In Sec. II B, we present iPINN-HL in detail, explaining its formulation, the integration of physics loss, and the neural network architecture employed. Section III shows the application of iPINN-HL across various scenarios, including one-dimensional spin chain systems, cross-resonance gate calibration, crosstalk identification, and parameter-drift compensation, with comparative analysis against DNN-HL. Finally, in Sec. IV, we summarize the findings, highlight the advantages of iPINN-HL, and discuss potential further developments.

II. PRELIMINARIES

A. Hamiltonian learning and maximum likelihood estimation

Let H be the Hamiltonian of an n -qubit system. It can be expressed as a linear combination of n -qubit Pauli strings:

$$H = \sum_{j=1}^{4^n} \theta_j P_j, \quad (1)$$

where $P_j \in \{I, \sigma_x, \sigma_y, \sigma_z\}^{\otimes n}$ and $\theta_j = \frac{1}{2^n} \text{Tr}(H P_j)$. Here, σ_x , σ_y , and σ_z are the Pauli matrices, and I is the identity matrix. In HL, the problem to learn the Hamiltonian of the system can be reduced to estimating a set of unknown parameters $\theta = \{\theta_1, \theta_2, \dots, \theta_{4^n}\}$ from experimental data. Data collection procedure of HL can be formulated as a query model defined in Ref. [26]. A query x consists of three components U , t and M , as illustrated in Fig. 1:

(1) *The state preparation unitary U .* The initial state $|\psi_0\rangle$ is specified by $|\psi_0\rangle = U|0\rangle^{\otimes n}$. In principle, $|\psi_0\rangle$ can be any unit vector in the Hilbert space, and this is generally assumed in most HL protocols. However, in the context of real device

calibration, greater attention must be paid to the constraints on the unitary U . For instance, in two-qubit gate calibration, it is not permissible to assume that the initial state $|\psi_0\rangle$ is entangled. In this case, the allowed U must have a tensor product form, i.e., $U = U_1 \otimes U_2$. Additionally, since the execution time of single-qubit gates is much shorter than the typical timescale of HL protocols, and the calibration of such gates is generally more accurate, it is often reasonable to assume that the state preparation process is not significantly affected by noise sources. Thus, we can consider the state preparation unitary U as ideal for practical purposes. For a practical example of device calibration, see Ref. [38].

(2) *The controlling parameter t .* The controlling parameter t refers to the time Hamiltonian H being applied to the system. In a typical HL protocol, t should take different values so that the dynamics during the time span can be accurately characterized. Algorithms exist to choose t systematically to reduce resources needed in HL [26]. However, in this work, we keep the controlling parameters equally spaced by Δt to examine the extrapolation ability of iPINN-HL. In other words, we aim to examine whether the embedded physics knowledge can increase the ability to infer unobserved data at other controlling parameter t and, in turn, reduce the resources required in HL.

(3) *The measurement operator M .* The measurement operator specifies the set of projective measurement bases $\{|\phi_m\rangle\}$. In this work, we assume that the final measurement bases are always fixed as the computational bases $\{|m\rangle\}$; therefore, $M = \sum_m |\phi_m\rangle\langle m|$. Similar to the constraint on the state preparation unitary U in two-qubit gate calibration, $|\phi_m\rangle$ should not be an entangled state.

We note that our definition of the measurement operator, $M = \sum_m |\phi_m\rangle\langle m|$, differs from the standard formalism, where projectors take the form $|\phi_m\rangle\langle\phi_m|$. In our construction, the computational basis $|m\rangle$ is fixed as the measurement basis, so that the outputs are always bit strings. This choice facilitates data processing in the neural network for iPINN-HL's loss term.

The system to be characterized receives a query

$$x = (U, t, M), \quad (2)$$

and outputs a single shot measurement result y , which is an n -bit string based on conditional probability defined as

$$p(y|x; \theta) = |\langle y|M e^{-iH(\theta)t} U |0\rangle^{\otimes n}|^2. \quad (3)$$

After collecting a dataset $D = \{(x_j, y_j)\}_{j=1}^N$, where j indexes each entry, we aim to estimate the unknown parameters θ . A natural approach is Maximum Likelihood Estimation (MLE), which produces an estimate $\hat{\theta}$ given by

$$\hat{\theta} = \arg \max_{\theta} \frac{1}{N} \sum_k \log_{10} p(y_k|x_k; \theta). \quad (4)$$

Accordingly, we define the loss function $\mathcal{L}(\theta|D)$ as

$$\mathcal{L}(\theta|D) = -\frac{1}{N} \sum_k \log_{10} p(y_k|x_k; \theta), \quad (5)$$

and by minimizing the loss function, we find the estimated parameter $\hat{\theta}$.

ALGORITHM 1. Deep-Neural-Network-based Hamiltonian Learning (DNN-HL).

Require: Dataset D , initial state $|\psi_0\rangle$, learning rate η , total iterations J , and ADAM optimizer

Ensure: Estimated Hamiltonian parameters $\hat{\theta}$

```

1: Initialize neural network weights  $w$  and Hamiltonian
   parameters  $\hat{\theta}$ 
2: for Epoch = 1 to  $J$  do
3:   Group dataset  $\tilde{D} = \{(U_k, t_k, M_k, \{y_k^{(s)}\}_{s=1}^{S_k})\}_{k=1}^N$ , where each
     group shares the same control parameters but includes multiple
     measurement outcomes;
4:   for Each measurement configuration  $(U_k, t_k, M_k, \{y_k^{(s)}\})$  in  $\tilde{D}$ 
     do
5:     Use the DNN to reconstruct quantum state  $|\phi_k\rangle$  from the
     measurement outcomes  $\{y_k^{(s)}\}_{s=1}^{S_k}$ 
6:     Compute the model-evolved state:
        $|\psi_k(\hat{\theta})\rangle = e^{-iH(\hat{\theta})t_k} U_k |\psi_0\rangle$ 
7:     Compute data loss:  $\mathcal{L}_{\text{data}}^{(k)} = \|\phi_k\rangle - |\psi_k(\hat{\theta})\rangle\|^2$ 
8:   end for
9:   Combine total loss:  $\mathcal{L}_{\text{total}} = \sum_k \mathcal{L}_{\text{data}}^{(k)}$ 
10:  Update  $w$  and  $\theta$  using ADAM optimizer with learning rate  $\eta$ 
11: end for
12: return Optimized Hamiltonian parameters  $\hat{\theta}$ 

```

The DNN-HL method is adapted from the neural-network-based quantum state tomography method proposed in Ref. [39], where a Restricted Boltzmann Machine (RBM) is used to reconstruct the quantum state from measurement data. In our work, we extend this idea to HL by learning the underlying Hamiltonian parameters from the reconstructed state. To align the model class with that of iPINN-HL and enable a fair comparison, we replace the RBM with a deep neural network. Although the architectures differ, both RBMs and DNNs are capable of representing complex distributions and learning nontrivial mappings from data under sufficient capacity. This makes the replacement legitimate in the context of benchmarking model performance for HL tasks. After the DNN is applied to reconstruct a set of state $\{|\phi_i\rangle\}$ for every controlling time parameter t_i by following the same routine demonstrated in Ref. [39], we minimize the discrepancy between the reconstructed state $|\phi_i\rangle$ and the estimated state $e^{-iH(\hat{\theta})t_i} |\psi_0\rangle$ at time parameter t_i . Hence, the loss function can be explicitly written as

$$\mathcal{L}_{\text{data}}^{\text{DNN}}(\hat{\theta}|D) = \sum_i \|\phi_i\rangle - e^{-iH(\hat{\theta})t_i} |\psi_0\rangle\|^2, \quad (6)$$

for all the initial states $|\psi_0\rangle$ generated by the state preparation unitary U .

The details of the DNN-HL algorithm are provided in Algorithm 1.

B. Inverse problem and PINNs

HL can be viewed as an inverse problem of a PDE—specifically, the Schrödinger equation:

$$i\hbar \frac{d|\Psi(t)\rangle}{dt} = H(\theta)|\Psi(t)\rangle, \quad (7)$$

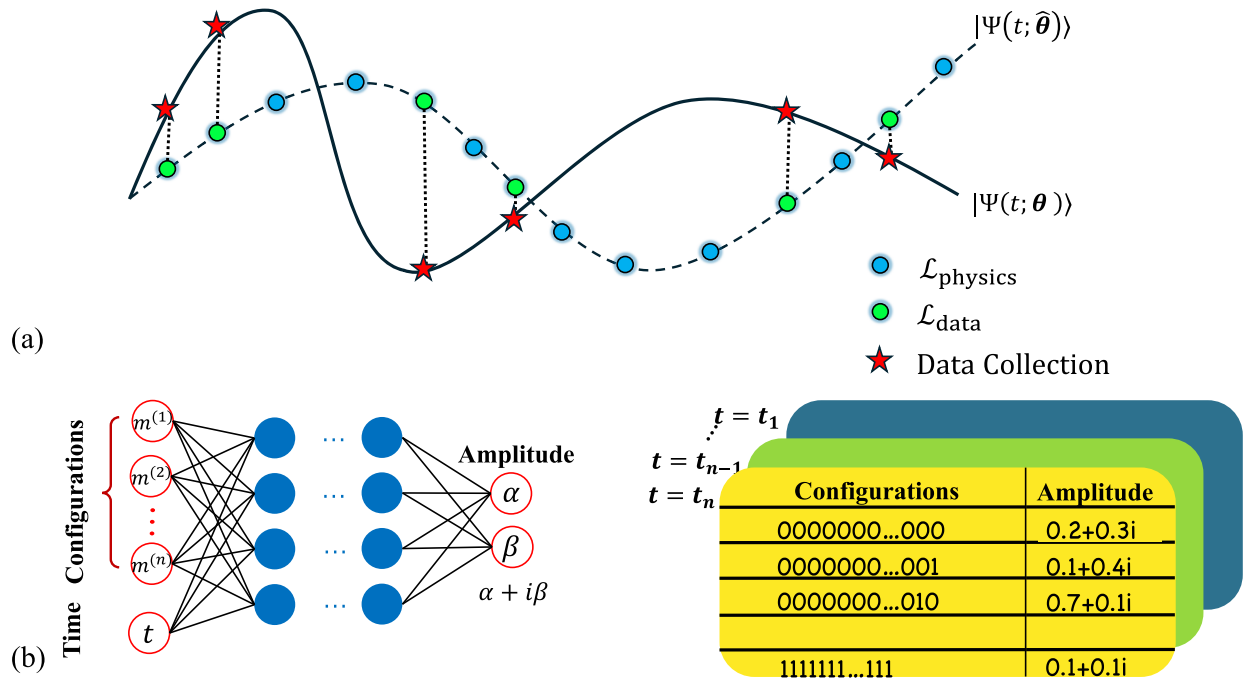


FIG. 2. (a) Visualization of PINNs for HL. The solid curve shows the true solution $|\Psi(t; \theta)\rangle$, while the dashed curve represents the estimated solution $|\Psi(t; \hat{\theta})\rangle$. Red stars mark data points, green dots indicate prediction errors, and blue dots enforce physical laws by ensuring consistency with the Schrödinger equation. (b) NNQS representation and its tabular form at different time points. The output nodes α and β of the neural network represent the real and imaginary parts, respectively, of the complex amplitude of the quantum state $\langle m|\Psi(t)\rangle$ at time t . The neural network efficiently captures the complex amplitudes of quantum states across various configurations, demonstrating high expressive capacity for representing many-body quantum systems. The table on the right shows the amplitudes of different configurations at discrete time steps $t_1, t_2, \dots, t_n$. Automatic differentiation of NNQS enables efficient computation of time derivatives of the quantum state, facilitating the enforcement of dynamical constraints of Schrödinger equation.

where the goal is to infer unknown Hamiltonian parameters from observed dynamics and measurements. For simplicity, we assume the state is pure in Eq. (7); therefore, the dynamics is unitary. This assumption applies to the remainder of this chapter as well as Secs. III A, III C, and III D. More complicated cases involving noises and therefore mixed states are considered in Sec. III B.

Figure 2(a) illustrates the fundamental concept of PINNs, supplemented with the Schrödinger equation. The solid curve represents the true solution of the system $|\Psi(t; \theta)\rangle$, where θ denotes system parameters. The dashed curve represents the estimated solution of the system $|\Psi(t; \hat{\theta})\rangle$ obtained under the proposed parameter estimation $\hat{\theta}$. The red stars indicate points where actual observed data are collected, and the model minimizes the discrepancy (shown by the vertical z lines) between predictions (green dots) and real-world measurements. The blue dots represent locations where the physical laws are enforced. At these points, the first derivative of the estimated state, $\frac{\partial |\Psi(t; \hat{\theta})\rangle}{\partial t}$, is computed alongside $|\Psi(t; \hat{\theta})\rangle$ to ensure that the system satisfies the Schrödinger equation in Eq. (7). Importantly, no data from the true solution are required at these points, as the physical laws themselves guide the learning process.

In order to utilize PINN to solve the HL problem, we first need to represent the quantum state $|\Psi(t)\rangle$ using a neural network. Accurately representing a quantum state can be computationally challenging due to the exponential scaling of the

state space with the number of particles n involved, illustrated in the right panel of Fig. 2(b), where one table corresponds to one time point and every table consists of 2^n entries. We address this challenge by parametrizing the quantum states with neural networks, denoted as Neural Network Quantum States (NNQSs). This method facilitates efficient approximations of states that would otherwise demand extensive computational resources [40,41]. In NNQS, the neural network acts as a parametrized wave function $\Psi_w(t, m)$, where w is the trainable weight in the network. The input is time t and the configuration m is the bit-string representation of the chosen computational basis $|m\rangle$, and its outputs correspond to the amplitudes of $|\Psi(t)\rangle$ in the chosen basis $|m\rangle$, $\langle m|\Psi(t)\rangle$. Therefore, NNQS allows for a continuous and differentiable approximation of the state across time. With the help of backpropagation, the NNQS can easily calculate and get the first-order differentiation of $|\Psi(t)\rangle$ with respect to time t , i.e., $\frac{d|\Psi(t)\rangle}{dt}$.

Implementing PINN, we randomly sample P points (which we call the number of *collocation points* [42] hereafter) in the time domain $(t_1, t_2, \dots, t_j, \dots, t_P)$. We incorporate another loss function (“physics loss”) in the training process (termed as $\mathcal{L}_{\text{physics}}$):

$$\mathcal{L}_{\text{physics}}(w, \theta) = \frac{1}{P} \sum_j \left| i\hbar \frac{d|\Psi(t_j)\rangle}{dt} - H(\theta)|\Psi(t_j)\rangle \right|^2. \quad (8)$$

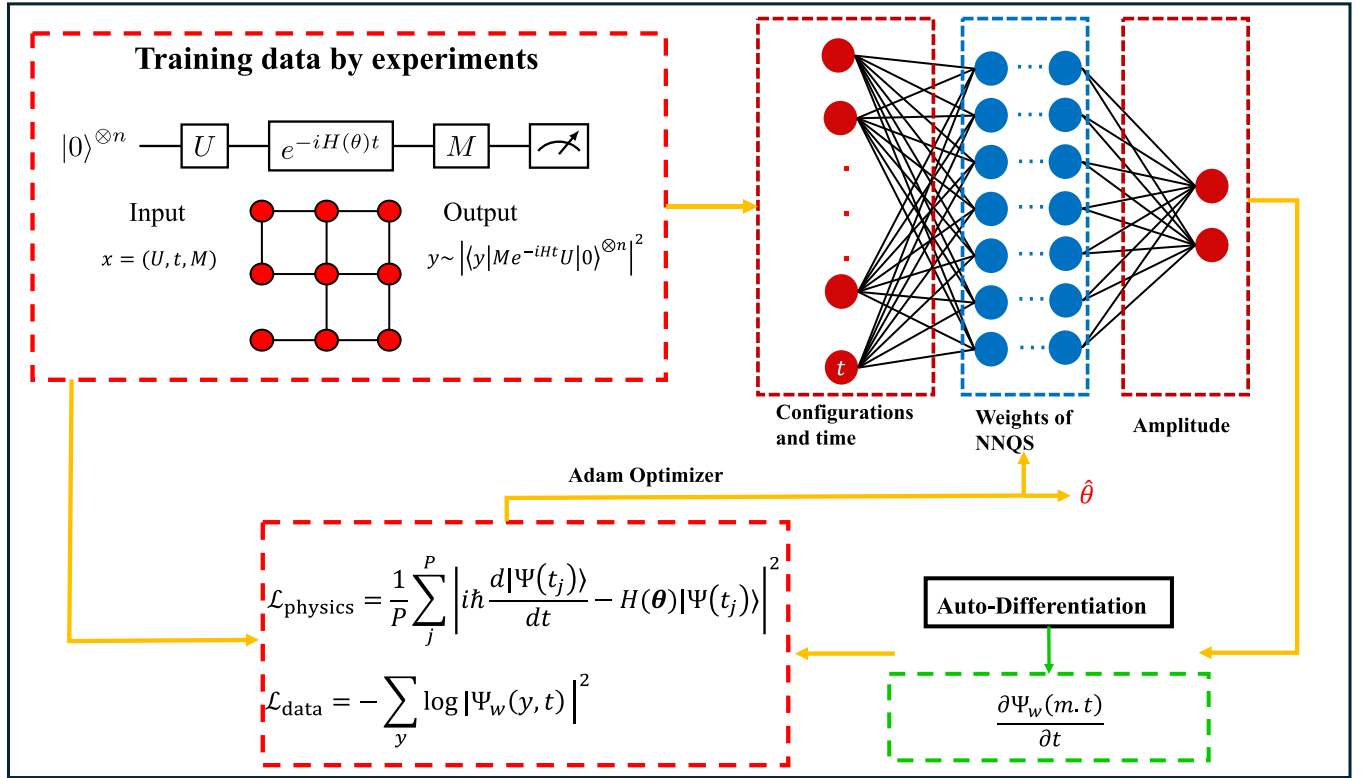


FIG. 3. Flowchart illustrating the use of PINNs for HL. Experimental data are generated by evolving an initial quantum state $|0\rangle^{\otimes n}$ under a unitary transformation U and time evolution $e^{-iH(\theta)t}$, followed by measurement M . The neural network takes configurations and time as inputs and outputs the amplitude of the quantum state, with its weights w and estimation $\hat{\theta}$ optimized using the Adam optimizer. The loss function combines a physical loss $\mathcal{L}_{\text{physics}}$, which ensures consistency with the Schrödinger equation, and a data loss $\mathcal{L}_{\text{data}}$, which minimizes discrepancies with experimental measurements. Automatic differentiation efficiently computes the time derivative $\partial \Psi_w(m, t)/\partial t$, facilitating the enforcement of physical constraints during training.

The full wave function $|\Psi(t_j)\rangle$ is assembled from the 2^n outputs of neural network,

$$|\Psi(t_j)\rangle = \sum_m (\alpha_m + i\beta_m) |m\rangle, \quad (9)$$

where α_m and β_m are the two outputs of neural network given input being configuration $|m\rangle$ and time t_j .

For every different initial quantum state $|\Psi^{(l)}(0)\rangle$, we use one NNQS $\Psi_w^{(l)}(t, m)$. The NNQS must satisfy the initial condition of Eq. (7). Hence, the loss function about initial condition termed as $\mathcal{L}_{\text{initial}}$ is also introduced in the training, defined as follows:

$$\mathcal{L}_{\text{initial}}(w^{(l)}, \theta|D) = \frac{1}{K} \sum_m |\langle m | \Psi^{(l)}(0) \rangle - \Psi_w^{(l)}(t=0, m)|^2, \quad (10)$$

where K is the total number of terms in the summation and $w^{(l)}$ is the weight of l th NNQS.

Together with a data loss function $\mathcal{L}_{\text{data}}$ defined similarly to Eq. (5),

$$\mathcal{L}_{\text{data}}(w, \theta|D) = -\frac{1}{N} \sum_k |\Psi_w(t_k, y_k)|^2, \quad (11)$$

where y_k is the observed bit string in the measurement result and t_k is the controlling parameter stored in the k th entry (x_k, y_k) of dataset D , we have the total loss function used in training PINN for HL:

$$\mathcal{L}_{\text{total}}(w, \theta|D) = \mathcal{L}_{\text{data}}(w, \theta|D) + \lambda_1 \mathcal{L}_{\text{physics}}(w, \theta) + \lambda_2 \mathcal{L}_{\text{initial}}(w, \theta|D), \quad (12)$$

where λ_1 and λ_2 are weighting factors. Note that $\mathcal{L}_{\text{physics}}(w, \theta)$ is not dependent on dataset D . Rather, it is incorporated to enforce that the evolution of NNQS satisfies the Schrödinger equation. In order to compare the accuracy of both algorithms under various settings, we use mean squared errors (MSE):

$$\text{MSE} = \frac{1}{K} \sum_i (\hat{\theta}_i - \theta_i)^2. \quad (13)$$

Here, θ_i denotes the true value of the i th parameter to be estimated in HL, which may arise from the Hamiltonian or the noise terms. Its estimate is denoted by $\hat{\theta}_i$, and K is the total number of parameters. We also report the relative mean squared error (relative MSE) as a performance metric:

$$\text{relative MSE} = \frac{\sum_i (\hat{\theta}_i - \theta_i)^2}{\sum_i \theta_i^2}. \quad (14)$$

ALGORITHM 2. Inverse Physics-Informed Neural Network for Hamiltonian learning.

Require: Dataset $D = \{(x_k, y_k)\}_{k=1}^N$ with $x_k = (U_k, t_k, M_k)$, initial conditions $|\Psi^{(l)}(0)\rangle$, total sampled points P , total HL protocol duration T , learning rate η , total iterations J and ADAM optimizer

Ensure: Hamiltonian parameters $\hat{\theta}$

- 1: Initialize parameters $\hat{\theta}$ and neural network weights $\{w^{(l)}\}$
- 2: **for** Epoch = 1 to J **do**
- 3: **for** Each initial condition $|\Psi^{(l)}(0)\rangle$ **do**
- 4: Sample P time points $\{t_j\}_{j=1}^P$ in time domain $[0, T]$
- 5: **for** each time point t_j **do**
- 6: **for** each configuration $m = 1$ to 2^n **do**
- 7: Compute NNQS amplitude $\Psi_w^{(l)}(t_j, m)$ and derivative $\frac{d\Psi_w^{(l)}(t_j, m)}{dt}$
- 8: **end for**
- 9: Assemble the full quantum state $|\Psi(t_j)\rangle = \sum_m \Psi_w^{(l)}(t_j, m)|m\rangle$ and its derivative $\frac{d|\Psi(t_j)\rangle}{dt} = \sum_m d\frac{\Psi_w^{(l)}(t_j, m)}{dt}|m\rangle$
- 10: Calculate **Physics Loss** according to Eq. (8)
- 11: Calculate **Initial Condition Loss** according to Eq. (10)
- 12: **end for**
- 13: **end for**
- 14: Calculate **Data Loss** for all data points in D according to Eq. (11)
- 15: Combine **Total Loss** according to Eq. (12)
- 16: Update θ and $\{w^{(l)}\}$ using ADAM optimizer
- 17: **end for**
- 18: **return** Optimized Hamiltonian parameters $\hat{\theta}$

We present iPINN-HL in Algorithm 2 and illustrate iPINN-HL in Fig. 3.

III. RESULTS

In this section, we present the results of iPINN-HL to various systems, including one-dimensional spin chains (Sec. III A), cross-resonance gates (Sec. III B), crosstalk effects (Sec. III C), and compensation of parameter drifts (Sec. III D).

A. One-dimensional spin chains

In this subsection, we apply iPINN-HL to estimate the Hamiltonian of a one-dimensional N -spin chain:

$$H_{\text{spin}} = \sum_{\langle i, j \rangle} J_{ij} \sigma_z^{(i)} \sigma_z^{(j)} + \sum_i \omega_i \sigma_z^{(i)}, \quad (15)$$

where $1 \leq i, j \leq N$ label the spins and $\langle i, j \rangle$ indicates the nearest neighbor. The interaction strength (J_{ij}) and local external field (ω_i) are parameters to be reconstructed. We further implement the periodic boundary condition so that $\langle N, 1 \rangle$ is a pair of nearest neighbors. We conduct two numerical experiments to benchmark iPINN-HL against the DNN-HL method. In the first experiment, we examine the efficiency of iPINN-HL as the system size scales and as the number of parameters to be estimated increases. First, we hold the inter-

action strength and external field uniform across the system (i.e., $J_{ij} = J$ and $\omega_i = \omega$ for all i and nearest-neighbor spins considered) and tune the number of spins N in the system to compare the two algorithms. The results are shown in Fig. 4. In Fig. 4, MSE, the key indicator of the accuracy of estimation, scales polynomially with the number of input queries $\mathcal{N}(\{x\})$ as $\mathcal{N}^{-\ell}(\{x\})$ for both iPINN-HL and DNN-HL and the scaling coefficients ℓ are shown in the legend. It is clearly seen that ℓ for iPINN-HL is very close to the Heisenberg limit ($\ell = 2$) when $N = 4, 7$ with only a slight decrease for $N = 10$. On the other hand, ℓ for DNN-HL are lower than the previous case by $\sim 10\%$ – 15% . This is a clear sign showing the advantage of iPINN-HL over DNN-HL.

Next, we allow J_{ij} and ω_i to take distinct values while respecting the periodic boundary condition. Namely,

$$(\hat{T}^s)^{-1} H \hat{T}^s = H, \quad (16)$$

where T is the translation operator mapping i to $i + 1$ or N to 1, while s is a parameter indicating the extent to which the translational symmetry is implemented. In this section, we fix $N = 8$ and the cases with $s = 1, 2, 4$ are exemplified in Fig. 5. One notes that different J_{ij} values as indicated in Fig. 5 imply the complexity of HL, with the case of $s = 4$ having the maximal complexity with eight distinct parameters (J_1 through J_4 and ω_1 through ω_4), the case of $s = 1$ the minimal complexity with two distinct parameters (J_1 and ω_1), and the case of $s = 2$ in between with four distinct parameters ($J_{1,2}$ and $\omega_{1,2}$).

Again, we calculate MSE as functions of number of queries $x = (U, t, M)$, and the results are shown in Fig. 6 with the fitted scaling coefficient ℓ indicated in the legend. We see that ℓ has values between 1.981 and 1.997 for iPINN-HL, which is higher than the results for DNN-HL (which value between 1.662 and 1.888). We further note that for $s = 1$, ℓ for iPINN-HL is about 5% higher than its counterpart DNN-HL, while this percentage increases to 19% for $s = 4$. This result suggests that iPINN-HL is more efficient than DNN-HL in the cases where the Hamiltonian has higher complexity. In the above study, we have shown that by incorporating physics loss, the burden of massive data collection task can be alleviated because *priori* physics knowledge is embedded in the learning process. The *physics loss* $\mathcal{L}_{\text{physics}}$ enforces that NNQS must satisfy the Schrödinger equation at P points as shown in Eq. (8). In the second numerical experiment, we evaluate how $\mathcal{L}_{\text{physics}}$ enhances algorithmic efficiency by analyzing the relationship between MSE and the number of queries $\mathcal{N}(\{x\})$. In the remainder of this subsection, we take $N = 8, s = 4$ with $J_1 = 1, J_2 = 0.5, J_3 = 2, J_4 = 1.5$ and $\omega_1 = 1, \omega_2 = 0.5, \omega_3 = 1.5, \omega_4 = 2$. We fix the number of (U, M) and vary the number of sampled points along the evolution trajectory of quantum state by changing the difference Δt in the controlling parameter t . A smaller Δt implies more data points in t , while a larger Δt suggests otherwise. The result can be found in Tables I and II, respectively, for iPINN-HL and DNN-HL. Overall, the MSE values in Table I are at least one order of magnitude smaller than those in Table II, confirming the superiority of iPINN-HL method. For iPINN-HL, reducing Δt from 2×10^{-1} to 2×10^{-2} decreases the MSE by a factor of 3–4. However, further reducing Δt to 2×10^{-3} yields only a marginal improvement (less than

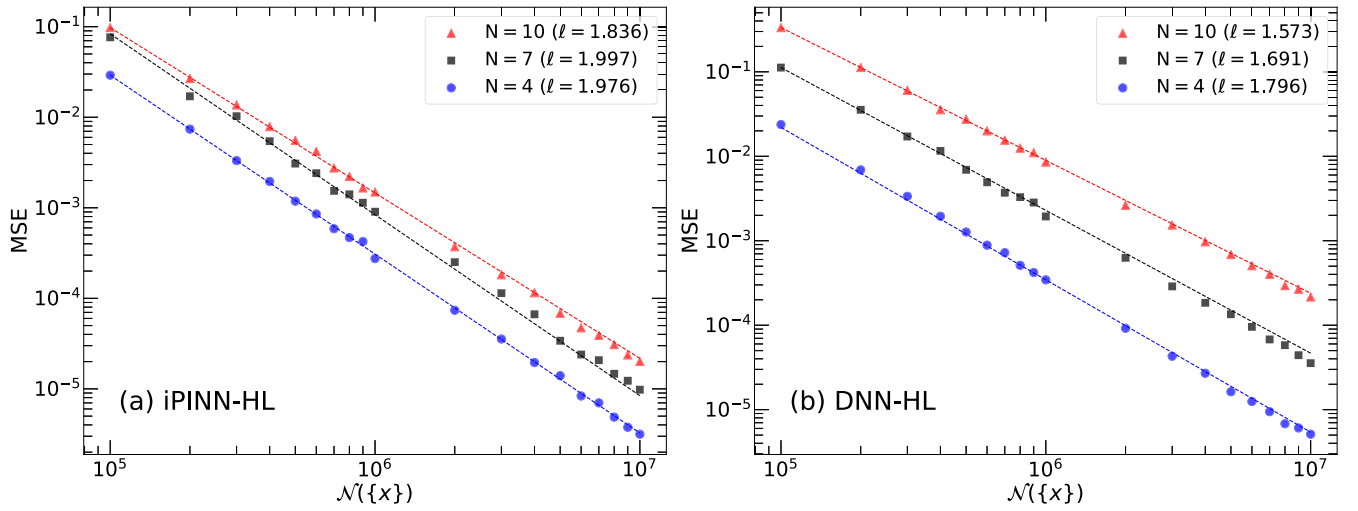


FIG. 4. The scaling plot of MSE of iPINN and DNN-HL with respect to the number of entries in the dataset D in panels (a) and (b), respectively, when the numbers of spins in one-dimensional spin chain are 4, 7 and 10. Here, we set $J = 1$, $\omega = 0.5$, and the number of collocation points $P = 50$.

a factor of 1.5). This suggests the existence of an optimal sampling density—beyond which additional queries provide diminishing returns in reconstruction accuracy for the HL problem. On the other hand, this phenomenon is not observed in the DNN-HL results (Table II), where the MSE shows no consistent improvement with decreasing Δt . This insensitivity to temporal resolution suggests that DNN-HL lacks the physics-informed regularization to achieve convergence with optimal sampling efficiency. The ability of iPINN-HL to extract meaningful physical constraints from sparser data compared to conventional deep learning approaches is therefore a key advantage.

Furthermore, we examine the performance of iPINN-HL when the number of collocation points P is varied. With larger P , the NNQS must satisfy the Schrödinger equation at more points and receives greater penalty from $\mathcal{L}_{\text{physics}}$ if it deviates from the prediction from the Schrödinger equation.

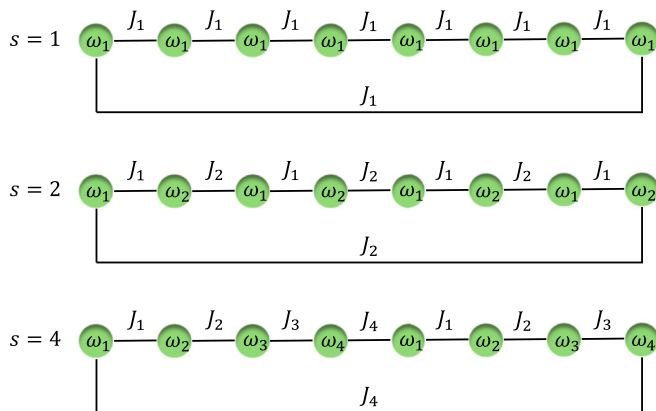


FIG. 5. Schematic representation of a one-dimensional spin chain system with translational symmetry parameters $s \in \{1, 2, 4\}$, showing the corresponding patterns of interaction strengths and local external field distributions.

The result can be found in Fig. 7. Figure 7 reveals that the MSE initially follows a polynomial decay with increasing P , where the scaling exponent grows with the number of queries $\mathcal{N}(\{x\})$. This demonstrates a synergistic relationship between data availability and the efficacy of the physics-informed loss $\mathcal{L}_{\text{physics}}$. Notably, when P exceeds a critical threshold ($P \approx 100$), the MSE saturates—particularly for small $\mathcal{N}(\{x\})$ —indicating a fundamental limit to physics-based enhancement without sufficient supporting data. These results highlight the complementary roles of data-driven and physics-driven learning: while $\mathcal{L}_{\text{physics}}$ significantly reduces the demand for extensive training data (compared to purely data-driven methods like DNN-HL), optimal performance requires a balanced integration of both $\mathcal{L}_{\text{data}}$ and $\mathcal{L}_{\text{physics}}$ components. This hybrid approach enables high-accuracy solutions with substantially fewer data points than conventional neural networks. To further elucidate the MSE saturation observed in Fig. 7, we consider an extremal case of no observational data ($\mathcal{N}(\{x\}) = 0$); minimizing $\mathcal{L}_{\text{physics}}$ alone permits convergence to any possible NNQS that satisfies the Schrödinger equation under some Hamiltonian, yielding a zero physics loss while the estimate $\hat{\theta}$ can take any value. Hence, physics loss alone provides no contribution without data loss. Increasing P beyond a threshold provides no further constraint without data, leading

TABLE I. The accuracy of estimation MSE of iPINN-HL in relation to differences Δt of controlling parameter t in data collection under fixed number of $\#(U, M)\#(U, M)$. Here, we set $P = 50$ and $N = 8$, $s = 4$ with $J_1 = 1$, $J_2 = 0.5$, $J_3 = 2$, $J_4 = 1.5$ and $\omega_1 = 1$, $\omega_2 = 0.5$, $\omega_3 = 1.5$, $\omega_4 = 2$.

$\#(U, M)$	$\Delta t = 2 \times 10^{-1}$	$\Delta t = 2 \times 10^{-2}$	$\Delta t = 2 \times 10^{-3}$
1×10^4	7.6636×10^{-2}	1.4119×10^{-2}	1.2756×10^{-2}
1×10^5	9.0494×10^{-4}	2.9375×10^{-4}	2.0281×10^{-4}
1×10^6	9.7906×10^{-6}	3.4510×10^{-6}	2.9611×10^{-6}

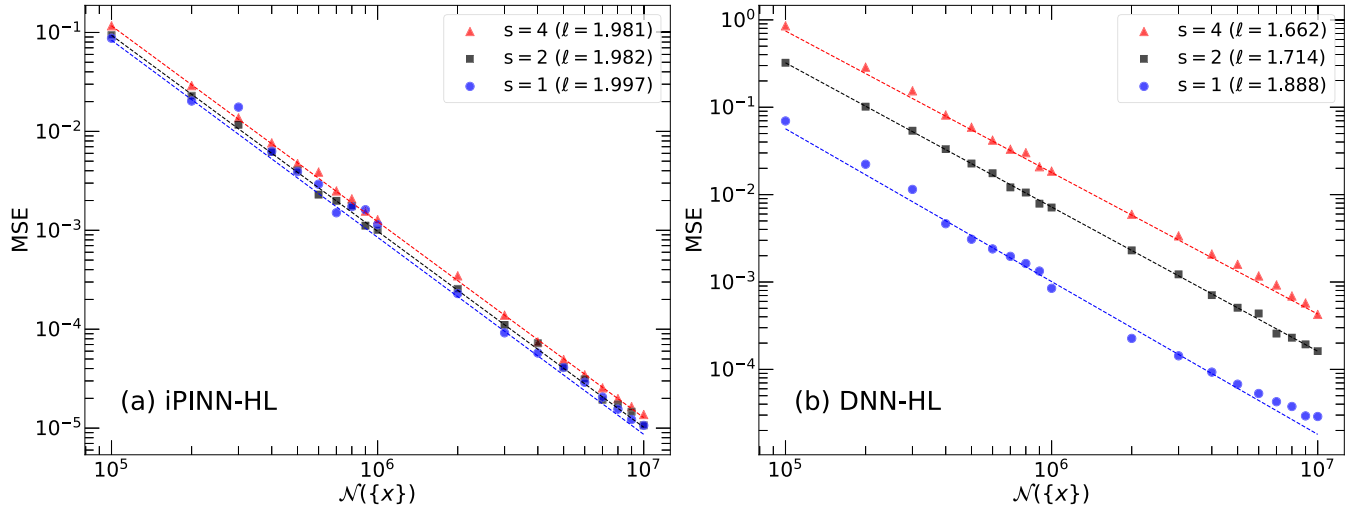


FIG. 6. MSE scaling with the number of queries $\mathcal{N}(\{x\})$ for translation symmetry parameters $s = 1, 2, 4$ in a one-dimensional spin chain, calculated from (a) iPINN-HL and (b) DNN-HL methods. Here, the number of collocation points P is set as 50. For $s = 1$, we set $J = 1$ and $\omega = 0.5$. For $s = 2$, $J_1 = 1, J_2 = 0.5$ and $\omega_1 = 1, \omega_2 = 0.5$. For $s = 4$, $J_1 = 1, J_2 = 0.5, J_3 = 2, J_4 = 1.5$ and $\omega_1 = 1, \omega_2 = 0.5, \omega_3 = 1.5, \omega_4 = 2$.

to such saturation. Similar phenomenon is also observed when solving the motion of a gravity pendulum using framework of PINN [43].

B. Cross-resonance gates

We employ iPINN-HL to reconstruct and calibrate the Hamiltonian of a quantum system implementing a CR gate. This challenging learning task is complicated by the inherent noise characteristics of multiqubit gates.

The CR gate serves as a fundamental two-qubit entangling gate in fixed-frequency superconducting transmon architectures, requiring only microwave control. Through appropriate pulse sequences, the CR gate can be converted into a locally equivalent CNOT gate, making it particularly valuable for quantum circuit compilation. The system Hamiltonian governing the CR gate dynamics is given by

$$H_{\text{CR}} = \frac{\sigma_z \otimes A}{2} + \frac{I \otimes B}{2}, \quad (17)$$

with $A = c_{zi}I + c_{zx}\sigma_x + c_{zy}\sigma_y + c_{zz}\sigma_z$ and $B = c_{ix}\sigma_x + c_{iy}\sigma_y + c_{iz}\sigma_z$, where σ_x, σ_y , and σ_z are the Pauli matrices and I is a 2×2 identity matrix. In this sense, HL for the CR gate reduces to estimating the parameter vector $\mathbf{c} = \{c_{zi}, c_{zx}, c_{zy}, c_{zz}, c_{ix}, c_{iy}, c_{iz}\}$. Importantly, we emphasize

TABLE II. The accuracy of estimation MSE of DNN-HL in relation to differences Δt of controlling parameter t in data collection under fixed number of $\#(U, M)\#(U, M)$. Here, we set $N = 8, s = 4$ with $J_1 = 1, J_2 = 0.5, J_3 = 2, J_4 = 1.5$ and $\omega_1 = 1, \omega_2 = 0.5, \omega_3 = 1.5, \omega_4 = 2$.

$\#(U, M)$	$\Delta t = 2 \times 10^{-1}$	$\Delta t = 2 \times 10^{-2}$	$\Delta t = 2 \times 10^{-3}$
1×10^4	3.1211×10^{-1}	3.0561×10^{-1}	2.7148×10^{-1}
1×10^5	6.8932×10^{-3}	6.3691×10^{-3}	5.4221×10^{-3}
1×10^6	1.4892×10^{-4}	1.0104×10^{-4}	9.0231×10^{-5}

that throughout this section and Sec. III C, our protocol operates under two key constraints: (1) No preparation of entangled states is permitted, and (2) measurements are restricted to product-state projections only. These operational restrictions reflect realistic limitations in many superconducting quantum processors while still enabling accurate Hamiltonian characterization.

The calibration of CR gates is particularly susceptible to noise in realistic quantum systems. In this work, we focus on two dominant noise sources: readout noise and decoherence, which represent the most significant limitations for Hamiltonian parameter estimation.

(1) *Readout noise.* Readout noise introduces errors during quantum state measurement, corrupting experimental data and

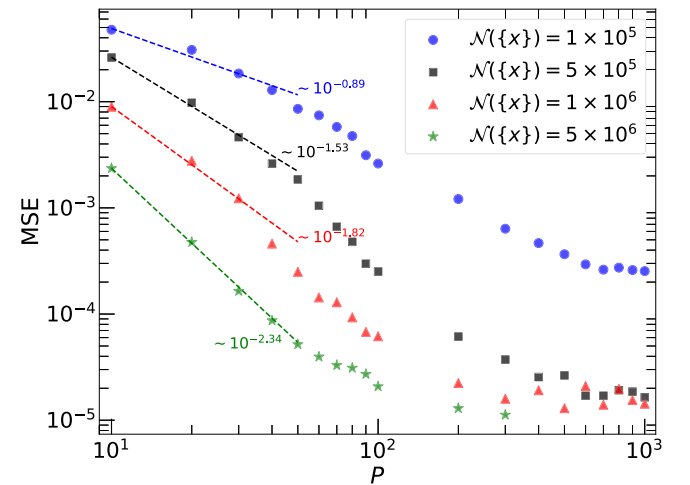


FIG. 7. Scaling of the MSE with the number of collocation points P for different number of queries $\mathcal{N}(\{x\})$. Dashed lines show fitted slopes of the initial polynomial decay regime with exponents indicated. Here, we set $P = 50$ and take $N = 8, s = 4$ with $J_1 = 1, J_2 = 0.5, J_3 = 2, J_4 = 1.5$ and $\omega_1 = 1, \omega_2 = 0.5, \omega_3 = 1.5, \omega_4 = 2$.

biasing the inferred Hamiltonian parameters. This directly affects the reliability of gate calibration.

The readout line of a qubit measurement result operates as a classical channel, where bit-flip errors can occur. Specifically, the true measurement outcome $y \in \{0, 1\}$ may be flipped. We denote the noisy measurement outcome observed in the experiment as $\tilde{y}$. The readout noise can be modeled by the conditional probability $p_{\tilde{y}|y}(\tilde{y}|y)$. In the absence of readout noise, we always have $p_{\tilde{y}|y}(\tilde{y} = y|y) = 1$. Therefore, the probability of obtaining the noisy outcome $\tilde{y}_k$ in the k th entry of the dataset D is given by

$$p(\tilde{y}_{ki}|x_k, \theta) = p_{\tilde{y}_{ki}|y_{ki}}(\tilde{y}_{ki}|\tilde{y}_{ki})p(y_{ki}|x_k, \theta) + p_{\tilde{y}_{ki}|y_{ki}}(\tilde{y}_{ki}|1 - \tilde{y}_{ki})p(y_{ki}|x_k, \theta), \quad (18)$$

where y_{ki} and $\tilde{y}_{ki}$ denote the true and noisy measurement outcomes, respectively, of the i th qubit in the k th entry of the dataset D , and $p(y|x, \theta)$ is defined in Eq. (5). In this study, we assume that the bit-flip channel is symmetric, i.e., $p_{\tilde{y}|y}(1|0) = p_{\tilde{y}|y}(0|1)$, and that the bit-flip channel is identical for all qubits in the system for simplicity.

(2) *Decoherence*. Decoherence degrades quantum coherence, leading to deviations from ideal unitary dynamics $U(t; \theta) = e^{-iH(\theta)t}$ and a reduction in gate fidelity. Here, we modeled it as a completely depolarized channel:

$$\mathcal{E}(\rho(t)) = (1 - p_d(t))\rho(t) + p_d(t)\frac{\mathcal{I}}{2^n}, \quad (19)$$

where $\rho(t) = U(t; \theta)\rho(0)U^\dagger(t; \theta)$ and $\mathcal{I}$ is a 4×4 matrix. To first order, the occurrence of depolarization events can be modeled as a Poisson process with rate μ :

$$p_d(t) = \exp\left(-\frac{t - t_0}{\mu}\right), \quad (20)$$

where t_0 is the starting time of the experiment. Therefore, identifying and quantifying decoherence is equivalent to estimating the Poisson rate μ . Same model is used in Ref. [26]. This approach approximates a Lindblad master equation with strong dephasing, mimicking rapid decoherence toward a maximally mixed state, and we include a full derivation in Appendix B to justify this approximation. Using this model for decoherence, we avoid making modification to the neural network to represent density matrix, which is harder to optimize when system scales up.

To summarize, the calibration of CR gate in the presence of noise is reduced to estimate the vector $\mathbf{c}$, bit-flip rate $p_{\tilde{y}|y}(\tilde{y}|y)$, and Poisson rate μ from the experiment data.

The relationship between estimation accuracy, measured by MSE, and the number of queries is illustrated in Fig. 8. It is evident from the figure that iPINN-HL consistently outperforms DNN-HL across both noiseless and noisy conditions. Notably, the MSE for iPINN-HL declines more rapidly with increasing query numbers, demonstrating its superior scalability and robustness. Noise sources, particularly readout noise, have a pronounced impact on the calibration process, as seen by the more gradual decline and greater spread in the DNN-HL curves compared to the iPINN-HL curves. Moreover, the presence of noise also affects training stability, as reflected in the less smooth scaling behavior of MSE for DNN-HL. These results confirm that iPINN-HL offers greater resilience against

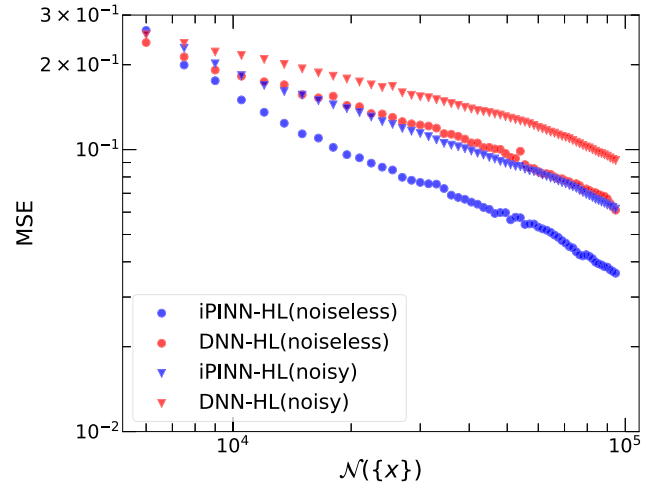


FIG. 8. Calibration accuracy of CR gate MSE as a function of the number of queries x in the dataset D , comparing iPINN-HL and DNN-HL methods under both noiseless and noisy conditions. Here, we set the parameter vector $\mathbf{c} = \{c_{zi}, c_{zx}, c_{zy}, c_{zz}, c_{ix}, c_{iy}, c_{iz}\} = \{0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.1\}$, Poisson rate $\mu = 5$, and the bit-flip error rate as 0.995 in the readout noise.

noise and is more efficient in terms of query usage compared to the traditional DNN-HL approach. The final relative MSE for the two algorithms under noisy and noiseless setting is shown in TABLE III.

C. Crosstalks

In this subsection, we focus on the HL problem for crosstalk effects in multiqubit systems. Crosstalk refers to unintended interactions between qubits in a quantum system that occur due to imperfections in hardware or control signals. For a system with N qubits, the general crosstalk effect can be modeled as

$$H_{\text{crosstalk}} = \sum_{i < j} \eta_{ij} (\sigma_x^{(i)} \sigma_x^{(j)} + \sigma_y^{(i)} \sigma_y^{(j)}) + \sum_{i < j} \epsilon_{ij} \sigma_z^{(i)} \sigma_z^{(j)} + \sum_i \delta_i \sigma_z^{(i)}. \quad (21)$$

The first term is caused by unintended XY-type interactions, where qubits influence each other through off-diagonal terms in the Hamiltonian. The second term is ZZ-type interactions, which can arise from dephasing-induced crosstalk when two qubits are affected by shared environmental noise sources or from residual coupling between two qubits. The last term is sometimes called “classical crosstalk” [44] since it is merely a shift in the qubit frequency and not qubit-qubit interaction. This can occur when qubits are not sufficiently detuned from

TABLE III. Relative MSE for iPINN-HL and DNN-HL under noiseless noisy settings, $\mathcal{N}(\{x\}) = 10^5$.

Method	iPINN-HL	DNN-HL
noiseless	1.1×10^{-2}	1.9×10^{-2}
noisy	1.8×10^{-2}	2.8×10^{-2}

TABLE IV. The table of crosstalk coefficients in the system with XY -type crosstalk (η_{ij}) shown in the left panel and ZZ -type (ϵ_{ij}) crosstalk in the right panel.

	Qubit 1	Qubit 2	Qubit 3	Qubit 4
Qubit 1	0.00	0.59	0.46	0.40
Qubit 2	0.59	0.00	0.35	0.48
Qubit 3	0.46	0.35	0.00	0.54
Qubit 4	0.40	0.48	0.54	0.00
	Qubit 1	Qubit 2	Qubit 3	Qubit 4
Qubit 1	0.00	0.25	0.37	0.23
Qubit 2	0.25	0.00	0.56	0.67
Qubit 3	0.37	0.56	0.00	0.27
Qubit 4	0.23	0.67	0.27	0.00

one another, causing their frequencies to overlap and leading to interference [45]. This can also occur when there is drive-pulse crosstalk [46].

Since qubits can be more appropriately detuned from each other with advancing quantum technology and the drive-pulse crosstalk can be readily characterized by measuring Rabi frequency, we focus on quantum crosstalk in this study:

$$H_{\text{crosstalk}} = \sum_{i < j} \eta_{ij} (\sigma_x^{(i)} \sigma_x^{(j)} + \sigma_y^{(i)} \sigma_y^{(j)}) + \sum_{i < j} \epsilon_{ij} \sigma_z^{(i)} \sigma_z^{(j)}. \quad (22)$$

Furthermore, in our query $x = (U, t, M)$ to the system, we assume that the initial preparation unitary U is perfect, and projective measurement onto product state is also perfect because single-qubit gate can be fine calibrated using dynamical decoupling to cancel out the interaction in the process. Read-out error and decoherence are not included in the model for simplicity but can also be estimated in principle.

Here, we conduct a numerical study to estimate the crosstalk of a four-qubit array. We set the parameters η_{ij} of XY -type crosstalk and ϵ_{ij} of ZZ -crosstalk according to Table IV. The accuracy of the parameter estimation, measured by MSE, is presented in Fig. 9. The figure clearly illustrates that iPINN-HL consistently achieves a significantly lower MSE across all crosstalk parameters compared to DNN-HL. Specifically, for both XY - and ZZ -type crosstalk, the iPINN-HL method exhibits MSE values that are approximately 5–10 times smaller than those of DNN-HL. This superior performance holds consistently across different parameter initializations and system configurations, as seen from the stable, low values of the results from iPINN-HL in the left column. Additionally, the MSE of DNN-HL displays greater fluctuations, indicating its sensitivity to initialization and its lower robustness when faced with complex crosstalk effects. These results highlight iPINN-HL's substantial advantage in both accuracy and stability when characterizing multi-qubit systems affected by crosstalk, reinforcing its potential for practical deployment in quantum device calibration and control.

D. Compensation for parameter drifts

We extend iPINN-HL to address the problem of parameter drifting in quantum systems, where Hamiltonian parameters

change over time due to environmental fluctuations or system instabilities. External factors such as temperature fluctuations, electromagnetic noise, and mechanical vibrations can subtly alter the system dynamics. Internally, qubit decoherence, residual interactions, and imperfections in hardware calibration contribute to this drift. Additionally, aging of materials or components in the device can cause long-term changes in performance.

In this section, we employ both iPINN-HL and DNN-HL to capture the parameter shift and recalibrate the system, thereby compensating for the effects of parameter drift. After training the iPINN-HL and DNN-HL algorithms to convergence, we introduce a sudden change to the system parameters. Although such abrupt changes are uncommon in real experimental settings, they serve as an effective testbed to assess the adaptability and robustness of both methods. Following this parameter shift, we collect a new batch of data $\Delta\mathcal{N}(\{x\})$ to refine the parameter estimates. $\Delta\mathcal{N}(\{x\})$ contains 300 queries as shown in Fig. 1 and $x = (U, t, M)$ is randomly sampled. For both algorithms, the pretrained network parameters are retained as the initial model configuration, and the networks are further trained using the new dataset $\Delta\mathcal{N}(\{x\})$. This approach enables the models to efficiently adapt to the updated system dynamics by integrating the latest observational data.

This approach closely aligns with the online learning paradigm in ML [47], where models incrementally adapt as new data become available, rather than being retrained from scratch. This way, iPINN-HL benefits from its pretrained structure, enabling it to rapidly converge to the new parameter regime after a sudden change. In contrast, DNN-HL requires more iterations and data to stabilize its estimates, as it lacks the inherent incorporation of governing physical principles that facilitate efficient adaptation.

For this study, we use the following two-qubit Hamiltonian:

$$H = \omega_1 \sigma_z^{(1)} + \omega_2 \sigma_z^{(2)} + \varepsilon \sigma_z^{(1)} \sigma_z^{(2)}, \quad (23)$$

where initially, $\omega_1 = \omega_2 = 0.5$ and $\varepsilon = 1$. After the sudden change, the parameters shift to $\omega_1 = \omega_2 = 1.5$ and $\varepsilon = 2$. In the numerical experiment, the size of the newly collected queries $\Delta\mathcal{N}(\{x\})$ is 300. The results, shown in Fig. 10, indicate that MSE for iPINN-HL quickly converges to zero after just eight batches of $\Delta\mathcal{N}(\{x\})$, while the MSE for DNN-HL decreases much more slowly and remains above 1 even

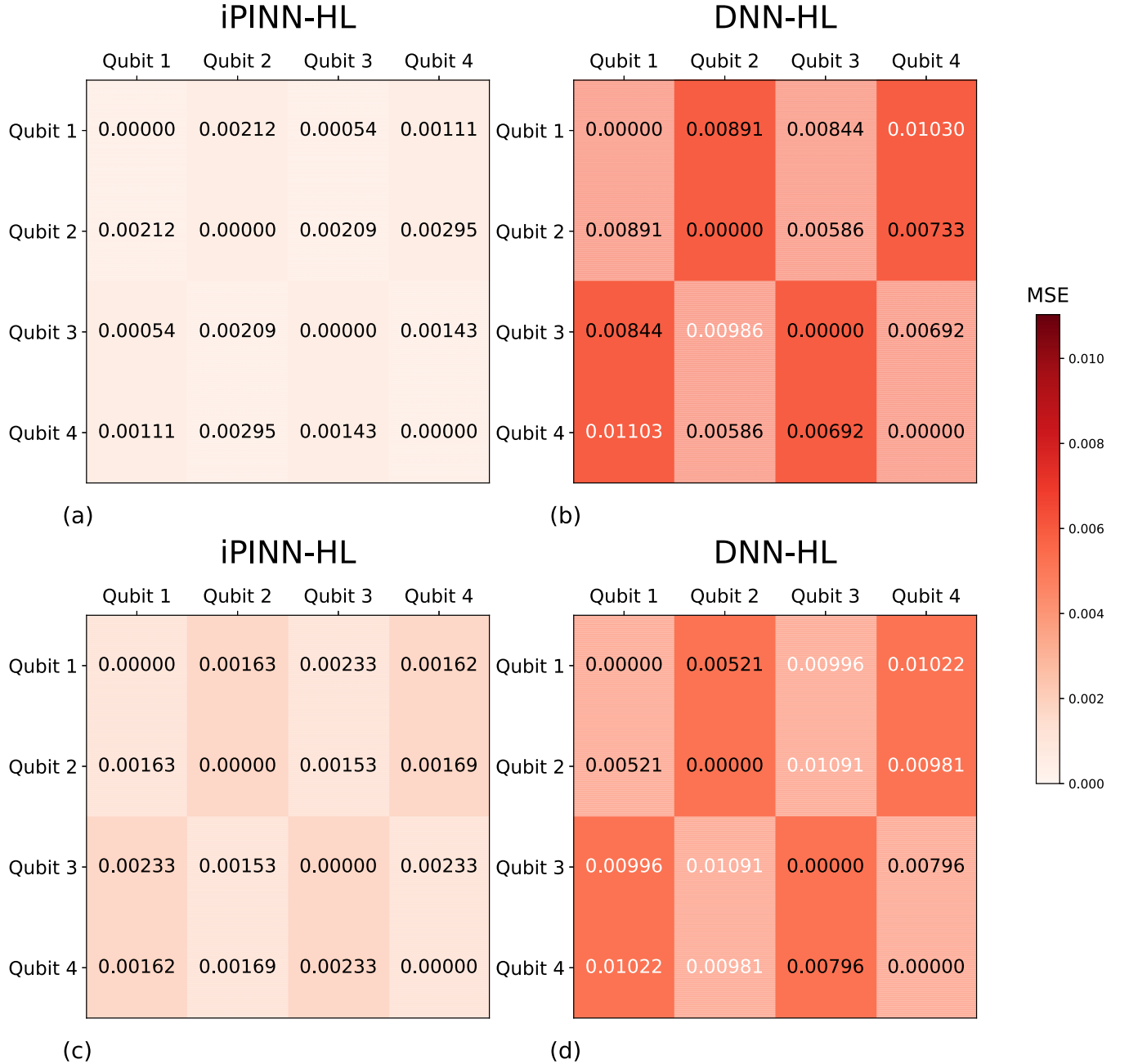


FIG. 9. The accuracy of estimation MSE (shown as pseudocolor scale) of coefficient for XY- and ZZ-type crosstalk using iPINN-HL in PANELS (a) and (c), and same for panels (b) and (d) using DNN-HL. The MSE is averaged based on multiple different initializations of algorithms. The crosstalk coefficients are set according to Table IV.

after ten batches. DNN-HL requires twice the amount of data to reach same level of accuracy again with iPINN-HL after parameter drift. This highlights the superior adaptability and convergence efficiency of iPINN-HL in response to abrupt system parameter changes.

IV. CONCLUSION AND OUTLOOK

In this study, we presented iPINN-HL, a physics-informed neural network framework for HL. Through comprehensive numerical experiments across multiple scenarios—including one-dimensional spin chain systems, cross-resonance gate

calibration, crosstalk identification, and parameter-drift compensation—we demonstrated its superior performance compared to DNN-HL. Our results, as detailed in Sec. III, show that iPINN-HL achieves higher estimation accuracy, enhanced robustness to noise, and greater adaptability in response to system changes.

The key advantage of iPINN-HL lies in its integration of physical principles into the learning process, enabling it to approach near-Heisenberg limit performance and maintain accuracy even in noisy environments. Its ability to adapt to dynamic system parameters makes it particularly suitable for real-time quantum system characterization and calibration.

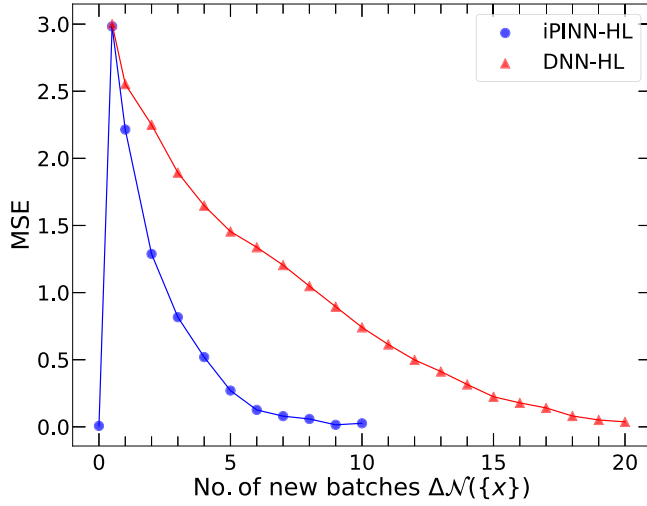


FIG. 10. The accuracy of estimation MSE of iPINN-HL and DNN-HL after a sudden change in the parameters of a two-qubit Hamiltonian defined in Eq. (23). Every new batch $\Delta\mathcal{N}(\{x\})$ contains 300 queries depicted in Fig. 1.

Moreover, iPINN-HL is highly compatible with advanced ML techniques. For instance, it can be integrated as a subroutine within other ML-based HL protocols, as demonstrated in prior works [23,24]. Additionally, iPINN-HL has the potential to be enhanced through active learning strategies [26], allowing for more efficient resource utilization while maintaining high accuracy. These features position iPINN-HL as a powerful and flexible tool for advancing HL in both research and practical quantum technology applications.

Future developments involving the integration of iPINN-HL with experimental quantum platforms hold considerable promise for enhancing the precision and reliability of quantum device calibration and control. Research directions include scaling iPINN-HL to larger and more complex quantum systems and devising hybrid strategies that combine classical optimization with quantum-assisted Hamiltonian learning. Extending iPINN-HL to open quantum systems with nonunitary dynamics also presents a compelling challenge, with the potential to bridge theoretical advances in Hamiltonian learning and practical applications in quantum technologies.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grants No. 11874312 and No. 12474489), the Research Grants Council of Hong Kong (CityU 11304920), Shenzhen Fundamental Research Program (Grant No. JCYJ20240813153139050), the Guangdong Provincial Quantum Science Strategic Initiative (Grants No. GDZX2203001 and No. GDZX2403001), and the Innovation Program for Quantum Science and Technology (Grant No. 2021ZD0302300). The calculations involved in this work were partially performed on the Tianhe Xingyi supercomputer at the National Supercomputer Center in Guangzhou, China. Open access was made possible with partial support from the Open Access Publishing Fund of the City University of Hong Kong.

DATA AVAILABILITY

The data that support the findings of this article, along with the source codes, are openly available [48]. Some training details of iPINN-HL and DNN-HL are also provided in Appendix C.

APPENDIX A: BASELINE COMPARISON

In this Appendix, we provide a comparison of several baseline methods for quantum state tomography, focusing on their scalability, computational cost, and accuracy. The methods considered are as follows:

(1) iPINN-HL (Physics-Informed Neural Networks for Hamiltonian Learning): The model proposed in this paper, where physics is incorporated into the learning process through a physics-informed loss term.

(2) DNN-HL (Deep Neural Network for Hamiltonian Learning): A traditional deep-neural-network approach to learning Hamiltonian parameters, without the inclusion of physics-informed constraints.

(3) RBM-HL (restricted Boltzmann machine for Hamiltonian learning): In the RBM-HL method, we replace the neural network architecture in DNN-HL with an RBM.

(4) MLE-HL (Maximum Likelihood Estimation): A traditional, non-ML approach used for quantum tomography and HL.

We confine this comparison to the one-dimensional spin chain model described in Sec. III A, as it enables a controlled evaluation of the scaling behavior, computational cost, and accuracy of each method under varying system sizes and numbers of parameters. This well-understood and computationally feasible model provides an ideal framework for assessing the core performance characteristics of the methods.

1. RBM-HL

RBM representation of quantum state. The RBM consists of a visible layer $\mathbf{v}$ and a hidden layer $\mathbf{h}$. We denote the visible layer input as $\mathbf{v}(t) = (m, t)$, where m is the bit-string representation of the computational basis state $|m\rangle$ and t is the time parameter. The hidden layer encodes latent variables that capture correlations in the quantum system. The energy function of the RBM is defined as

$$E(\mathbf{v}, \mathbf{h}; \mathbf{W}, \mathbf{b}, \mathbf{c}) = - \sum_i v_i c_i - \sum_j h_j b_j - \sum_{i,j} v_i w_{ij} h_j, \quad (\text{A1})$$

where $\mathbf{v} = (m_1, m_2, \dots, m_n, t)$ represents the visible units, $\mathbf{h} = (h_1, h_2, \dots, h_m)$ are the hidden units, w_{ij} is the weight connecting the i th visible unit and the j th hidden unit, and c_i and b_j are the biases for visible and hidden layers, respectively. To model a quantum state, the RBM defines a variational wave-function ansatz. The amplitude associated with a configuration $\mathbf{v}$ is

$$\Psi_{\omega}(\mathbf{v}) = \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h}; \mathbf{W}, \mathbf{b}, \mathbf{c})}, \quad (\text{A2})$$

where the parameters $\theta = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$ can be complex valued. The normalized quantum state is then given by

$$|\Psi_\omega\rangle = \frac{1}{\sqrt{\mathcal{N}}} \sum_{\omega} \Psi_\omega(\mathbf{v}) |\mathbf{v}\rangle, \quad (\text{A3})$$

with normalization constant

$$\mathcal{N} = \sum_{\mathbf{v}} |\Psi_\omega(\mathbf{v})|^2. \quad (\text{A4})$$

The probability of observing outcome $\mathbf{v}$ upon measurement in the computational basis follows the Born rule:

$$P(\mathbf{v}) = \frac{|\Psi_\omega(\mathbf{v})|^2}{\sum_{\mathbf{v}'} |\Psi_\omega(\mathbf{v}')|^2}. \quad (\text{A5})$$

Training the RBM consists of adjusting the weights and biases to minimize the difference between the model distribution and the measurement statistics. This is typically done using contrastive divergence (typically denoted as CD- k [49]) in cases where the target distribution is directly accessible from experimental measurement outcomes.

RBM training from quantum measurements. Given a set of measurement samples $\{\mathbf{v}^{(i)}\}$ drawn from the true quantum state, the RBM parameters $\omega = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$ are optimized such that the model probability distribution $P_\omega(\mathbf{v})$ matches the empirical statistics. The optimization objective is the log-likelihood

$$\mathcal{L}(\omega) = \sum_i \log_{10} P_\omega(\mathbf{v}^{(i)}). \quad (\text{A6})$$

Its gradient can be written as

$$\frac{\partial \mathcal{L}}{\partial \omega} = \mathbb{E}_{\text{data}} \left[\frac{\partial E}{\partial \omega} \right] - \mathbb{E}_{\text{model}} \left[\frac{\partial E}{\partial \omega} \right], \quad (\text{A7})$$

where

$$\mathbb{E}_{\text{data}}[f(\mathbf{v}, \mathbf{h})] = \frac{1}{N} \sum_{i=1}^N \sum_{\mathbf{h}} P_\omega(\mathbf{h}|\mathbf{v}^{(i)}) f(\mathbf{v}^{(i)}, \mathbf{h}), \quad (\text{A8})$$

$$\mathbb{E}_{\text{model}}[f(\mathbf{v}, \mathbf{h})] = \sum_{\mathbf{v}, \mathbf{h}} P_\omega(\mathbf{v}, \mathbf{h}) f(\mathbf{v}, \mathbf{h}). \quad (\text{A9})$$

The first term, known as the *positive phase*, is directly estimated from measurement data. The second term, known as the *negative phase*, requires sampling from the model distribution. The parameter updates then read

$$\Delta W = \eta(\langle \mathbf{v}\mathbf{h}^\top \rangle_{\text{data}} - \langle \mathbf{v}\mathbf{h}^\top \rangle_{\text{model}}), \quad (\text{A10})$$

$$\Delta \mathbf{c} = \eta(\langle \mathbf{v} \rangle_{\text{data}} - \langle \mathbf{v} \rangle_{\text{model}}), \quad (\text{A11})$$

$$\Delta \mathbf{b} = \eta(\langle \mathbf{h} \rangle_{\text{data}} - \langle \mathbf{h} \rangle_{\text{model}}), \quad (\text{A12})$$

with learning rate η . To approximate $\mathbb{E}_{\text{model}}$, one uses Gibbs sampling (i.e., CD- k). For binary RBMs, the conditional probabilities needed for Gibbs updates are

$$P(h_j = 1|\mathbf{v}) = \sigma(b_j + \mathbf{v}^\top \mathbf{W}_{\cdot j}), \quad (\text{A13})$$

$$P(v_i = 1|\mathbf{h}) = \sigma(c_i + \mathbf{W}_i \cdot \mathbf{h}), \quad (\text{A14})$$

where $\sigma(x) = 1/(1 + e^{-x})$. In CD- k , one initializes $\mathbf{v}$ with a data sample, alternates sampling $\mathbf{h}$ and $\mathbf{v}$ for k steps, and

uses the resulting samples to estimate the negative-phase expectations. Unlike standard DNNs, which rely on direct backpropagation through deterministic forward passes, RBM training requires sampling from the model distribution to estimate the negative-phase statistics. This involves iterative Gibbs sampling or its variants, which can be computationally demanding due to the need for repeated stochastic updates and Markov chain mixing. As a result, each training step of an RBM typically incurs significantly higher computational cost compared to DNNs, especially when scaling to large systems.

2. MLE-HL

MLE is a statistical method used to estimate the parameters of a probabilistic model based on observed data. The core idea of MLE is to find the parameter values that maximize the likelihood of the observed data under the estimate. For the task of HL, the likelihood function can be defined as follows:

$$\mathcal{L}(\hat{\theta}) = \sum_i |\langle y_i | M e^{-iH(\hat{\theta})t} U | 0 \rangle^{\otimes n}|^2. \quad (\text{A15})$$

The summation is taken over all the sampled bit strings obtained from the measurements. In simulation, the likelihood function is directly optimized using Broyden-Fletcher-Goldfarb-Shanno algorithm (BFGS).

3. Comparison result

The results of the benchmark presented in Fig. 11 provide a comprehensive comparison of the iPINN-HL against three baseline methods. Scaling coefficients indicate precision scaling with number of queries. Training times are reported on an NVIDIA RTX 4070 Ti Super GPU. Accuracy metrics (MSE and relative MSE) are computed for $\mathcal{N}(\{x\}) = 10^6$ queries, while other parameters are held the same as in Sec. III A and Appendix C. Across all examined system sizes ($N = 4, 7, 8, 10$) and translation symmetry levels ($s = 1, 2, 4$), iPINN-HL consistently achieves the highest scaling coefficient, approaching the Heisenberg limit [$\mathcal{O}(1/N^2)$]. This indicates superior precision scaling with data, where the estimation error decreases quadratically, enabling efficient learning even as the complexity of the quantum system increases. In contrast, DNN-HL and RBM-HL exhibit lower scaling coefficients ranging from 1.501 to 1.796, while MLE-HL yields the lowest scaling coefficients (1.403–1.621). Notably, iPINN-HL's scaling coefficient remains almost invariant across different s , demonstrating robustness to the number of Hamiltonian parameters, whereas the three other baselines show a clear degradation with increasing s . Furthermore, iPINN-HL outperforms the other methods in terms of estimation accuracy, exhibiting the lowest MSE and relative MSE for all configurations considered. For instance, at $N = 7$ and $s = 1$, iPINN-HL achieves an MSE of 1.0×10^{-3} and relative MSE of 1.8×10^{-3} , compared to 3.1×10^{-3} and 5.5×10^{-3} for DNN-HL, 9.0×10^{-3} and 1.6×10^{-2} for RBM-HL, and 1.5×10^{-2} and 2.7×10^{-2} for MLE-HL. This advantage stems from the integration of Schrödinger equation into the loss function, which constrains the solution space to physically consistent outcomes and enhances data efficiency. The baselines, lacking such physical priors, exhibit

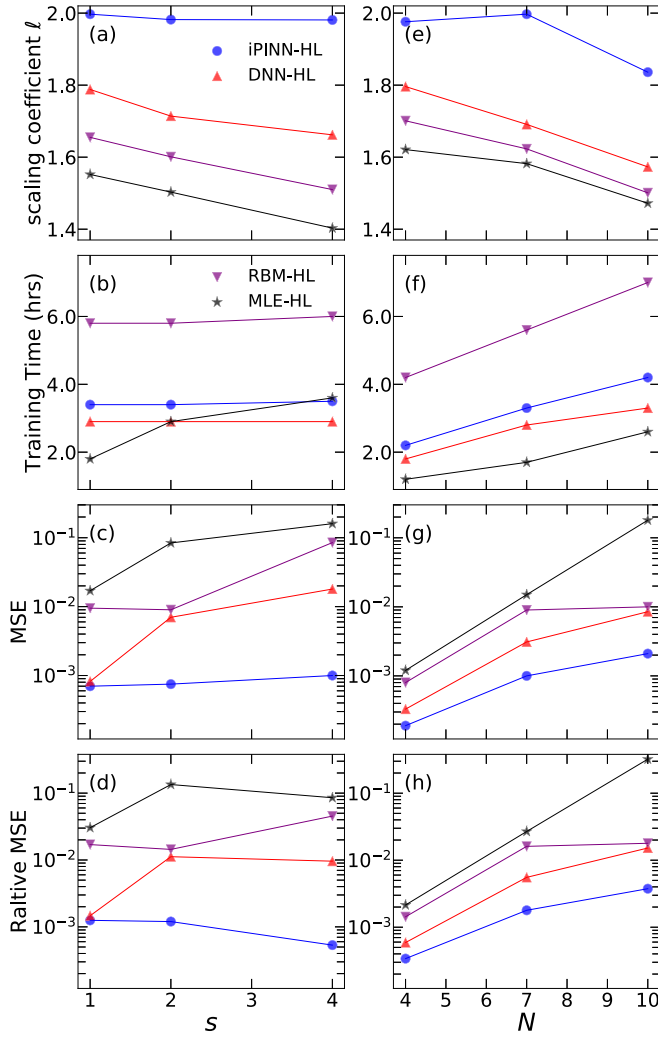


FIG. 11. Performance and computational cost of iPINN-HL, DNN-HL, RBM-HL, and MLE-HL for the one-dimensional spin chain model. The left column shares the same x axis, namely, the translational symmetry factor ($s = 1, 2, 4$) while $N = 8$; the right column shares the same x axis, i.e., the number of spins ($N = 4, 7, 10$) while $s = 1$. Each row shares the same y axis, showing (a, e) the scaling coefficient, (b, f) training hours, (c, g) MSE, and (d, h) relative MSE.

higher errors that increase more rapidly with N and s . While iPINN-HL demonstrates clear superiority in scaling and accuracy, it incurs higher training times compared to DNN-HL and MLE-HL, reflecting the additional computational overhead of incorporating physical constraints. Specifically, the training time for iPINN-HL is elevated relative to DNN-HL due to the evaluation and minimization of the physics loss over collocation points, which involves computing time derivatives of the quantum state and enforcing consistency with the Schrödinger equation at multiple sampled times. This adds complexity to each optimization iteration, resulting in times such as 3.3 h for $N = 7, s = 1$, versus 2.8 h for DNN-HL. Similarly, RBM-HL requires longer training owing to its sampling-based nature, where contrastive divergence or Gibbs sampling over the Hilbert space introduces stochastic overhead, leading to times like 5.6 h for $N = 7, s = 1$. ML

methods (iPINN-HL, DNN-HL, and RBM-HL) generally exceed the non-ML MLE-HL in training duration because they involve differentiation over network parameters in addition to the unknown Hamiltonian parameters, necessitating back-propagation through large models. For MLE-HL, the training time is dominated by repeated matrix exponentials, but BFGS optimization keeps it low because the differentiation only acts on the unknown Hamiltonian parameters without a neural network in the loop, e.g., 1.7 h for $N = 7, s = 4$. Nevertheless, the generalization capabilities of ML methods, particularly iPINN-HL, mitigate the high cost of data collection in the quantum regime. Traditional approaches like MLE-HL require dense sampling of the dynamics to achieve comparable accuracy, often demanding extensive experimental resources. In contrast, iPINN-HL leverages learned representations and physical priors to extrapolate beyond observed data, reducing the reliance on large datasets and enabling efficient characterization of quantum systems where measurements are expensive or limited. This trade-off, higher computational time for lower data requirements, positions iPINN-HL as a practical tool for advancing quantum technologies, where resource efficiency is paramount.

APPENDIX B: DERIVATION OF NOISE CHANNELS FROM THE LINDBLAD MASTER EQUATION

In this Appendix, we show the derivations that justify the approximation of solving Lindblad master equation by applying noise channels after unitary gates, in noisy quantum circuit simulations. The goal is to show how continuous-time dynamics can be approximated as discrete quantum channels in a circuit model. The Lindblad master equation governs the evolution of a quantum system's density matrix $\rho(t)$ under Markovian noise:

$$\frac{d\rho(t)}{dt} = -\frac{i}{\hbar}[H, \rho(t)] + \sum_k \mathcal{D}[L_k]\rho(t), \quad (\text{B1})$$

where H is the system Hamiltonian, $[H, \rho] = H\rho - \rho H$, and the dissipative term is

$$\mathcal{D}[L_k]\rho = L_k\rho L_k^\dagger - \frac{1}{2}\{L_k^\dagger L_k, \rho\}, \quad (\text{B2})$$

with $\{A, B\} = AB + BA$ and L_k as the Lindblad operators describing noise processes. We consider the evolution over a small time step Δt . The solution to the master equation is

$$\rho(t + \Delta t) = e^{\mathcal{L}\Delta t}\rho(t), \quad (\text{B3})$$

where the Liouvillian superoperator is

$$\mathcal{L}\rho = -\frac{i}{\hbar}[H, \rho] + \sum_k \left(L_k\rho L_k^\dagger - \frac{1}{2}\{L_k^\dagger L_k, \rho\} \right). \quad (\text{B4})$$

For small Δt , we approximate

$$e^{\mathcal{L}\Delta t} \approx 1 + \mathcal{L}\Delta t, \quad (\text{B5})$$

yielding

$$\rho(t + \Delta t) \approx \rho(t) + \mathcal{L}\rho(t)\Delta t. \quad (\text{B6})$$

TABLE V. Training details for iPINN-HL and DNN-HL. Training will be early stopped after 200 epochs of no validation loss improvement. Here, lr (learning rate) sets the step size of updates, β_1 (first moment decay) controls the smoothing of momentum, β_2 (second moment decay) regulates the smoothing of variance, and ϵ (stability constant) prevents division by zero.

Method	iPINN-HL	DNN-HL
Network structure	Feedforward NN, five layers (120 nodes each)	Feedforward NN, five layers (120 nodes each)
Activation function	SiLU [50]	Tanh
Optimizer	ADAM ($\text{lr} = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$) + BFGS	ADAM ($\text{lr} = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$)
Training epochs	3000–5000, early stopping	3000–5000, early stopping

Now, we split the Liouvillian into coherent and dissipative parts:

- (1) Coherent: $\mathcal{L}_{\text{coh}}\rho = -\frac{i}{\hbar}[H, \rho]$,
 - (2) Dissipative: $\mathcal{L}_{\text{dis}}\rho = \sum_k (L_k \rho L_k^\dagger - \frac{1}{2}\{L_k^\dagger L_k, \rho\})$.
- Using the Trotter-Suzuki approximation for small Δt ,

$$e^{\mathcal{L}\Delta t} \approx e^{\mathcal{L}_{\text{coh}}\Delta t} e^{\mathcal{L}_{\text{dis}}\Delta t} + O(\Delta t^2), \quad (\text{B7})$$

the coherent part gives unitary evolution:

$$e^{\mathcal{L}_{\text{coh}}\Delta t} \rho = e^{-iH\Delta t/\hbar} \rho e^{iH\Delta t/\hbar}. \quad (\text{B8})$$

Since typical quantum gate durations t_g are much shorter than the characteristic coherence times of the system (for superconducting qubits, $t_g \sim 10\text{--}100\text{ ns}$ while $T_{1,2} \sim 10\text{--}100\text{ }\mu\text{s}$), the probability of a decoherence event within a single gate is very small. This justifies treating the gate duration as a short-time step in the Trotter expansion. Thus, we identify $\Delta t = t_g$, so that the coherent unitary dynamics and the dissipative noise can be separated to first order. For a gate time $t_g = \Delta t$, the unitary gate is $U = e^{-iHt_g/\hbar}$, so

$$\rho \rightarrow U \rho U^\dagger. \quad (\text{B9})$$

The dissipative part is approximated as

$$e^{\mathcal{L}_{\text{dis}}t_g} \rho \approx \rho + t_g \sum_k \left(L_k \rho L_k^\dagger - \frac{1}{2}\{L_k^\dagger L_k, \rho\} \right). \quad (\text{B10})$$

This can be written as a quantum channel:

$$\rho \rightarrow \mathcal{E}(\rho) = \sum_k K_k \rho K_k^\dagger, \quad (\text{B11})$$

with Kraus operators:

- (1) $K_0 = I - \frac{t_g}{2} \sum_k L_k^\dagger L_k$,
- (2) $K_k = \sqrt{t_g} L_k$ for each L_k .

The Kraus operators satisfy $\sum_k K_k^\dagger K_k = I$ to first order, ensuring a trace-preserving map. In a quantum circuit, we approximate the evolution over a gate time t_g as a unitary gate

followed by a noise channel:

$$\rho \rightarrow \mathcal{E}(U \rho U^\dagger), \quad (\text{B12})$$

where $U = e^{-iHt_g/\hbar}$ and $\mathcal{E}$ is the noise channel derived from the dissipative terms over t_g . As an example, we consider a single qubit with depolarizing noise. The master equation includes

$$\frac{d\rho}{dt} = -\frac{i}{\hbar}[H, \rho] + \frac{\gamma}{4} \sum_{i=1}^3 \left(\sigma_i \rho \sigma_i - \frac{1}{2}\{\sigma_i^2, \rho\} \right), \quad (\text{B13})$$

where $\sigma_i = \{X, Y, Z\}$ and $\sigma_i^2 = I$. The dissipative term becomes

$$\mathcal{D}[\sigma_i]\rho = \sigma_i \rho \sigma_i - \rho. \quad (\text{B14})$$

Over gate time t_g , the noise channel is

$$\mathcal{E}_{\text{dis}}(\rho) = (1-p)\rho + \frac{p}{3} \sum_{i=1}^3 \sigma_i \rho \sigma_i, \quad (\text{B15})$$

with $p = 3\gamma t_g/4$. The Kraus operators are

- (1) $K_0 = \sqrt{1-p}I$,
- (2) $K_i = \sqrt{p/3}\sigma_i$ for $i = 1, 2, 3$.

The full evolution after a gate is

$$\rho \rightarrow (1-p)U \rho U^\dagger + \frac{p}{3} \sum_{i=1}^3 \sigma_i U \rho U^\dagger \sigma_i. \quad (\text{B16})$$

This isotropic depolarizing channel is equivalent to the heuristic model where, with probability p , the state is replaced by the maximally mixed state I/d . Indeed, the map can be written as $\mathcal{E}(\rho) = (1-p)\rho + pI/d$, which matches the action of the depolarizing channel derived from the Lindblad form. This matches the model used in Sec. III B. However, for non-Markovian noise or long gate times, the full master equation must be solved numerically.

APPENDIX C: TRAINING DETAILS OF IPINN-HL AND DNN-HL

In TABLE V, we present the training details of both iPINN-HL and DNN-HL presented in the main text.

[1] G. Dridi, K. Liu, and S. Guérin, Optimal robust quantum control by inverse geometric optimization, *Phys. Rev. Lett.* **125**, 250403 (2020).

[2] T. Chen, J.-Q. Hu, C. Zhang, and Z.-Y. Xue, Universal robust geometric quantum control via geometric trajectory correction, *Phys. Rev. Appl.* **22**, 014060 (2024).

- [3] H. Ball, M. J. Biercuk, A. R. Carvalho, J. Chen, M. Hush, L. A. De Castro, L. Li, P. J. Liebermann, H. J. Slatyer, C. Edmunds *et al.*, Software tools for quantum control: Improving quantum computer performance through noise and error suppression, *Quantum Sci. Technol.* **6**, 044011 (2021).
- [4] V. Sivak, A. Eickbusch, H. Liu, B. Royer, I. Tsoutsios, and M. Devoret, Model-free quantum control with reinforcement learning, *Phys. Rev. X* **12**, 011059 (2022).
- [5] M. Mohan, R. de Keijzer, and S. Kokkelmans, Robust control and optimal Rydberg states for neutral atom two-qubit gates, *Phys. Rev. Res.* **5**, 033052 (2023).
- [6] R. Acharya, L. Aghababaie-Beni, I. Aleiner, T. I. Andersen, M. Ansmann, F. Arute, K. Arya, A. Asfaw, N. Astrakhantsev, J. Atalaya *et al.*, Quantum error correction below the surface code threshold, *Nature (London)* **638**, 920 (2025).
- [7] S. Bravyi, A. W. Cross, J. M. Gambetta, D. Maslov, P. Rall, and T. J. Yoder, High-threshold and low-overhead fault-tolerant quantum memory, *Nature (London)* **627**, 778 (2024).
- [8] J.-S. Chen, E. Nielsen, M. Ebert, V. Inlek, K. Wright, V. Chaplin, A. Maksymov, E. Pérez, A. Poudel, P. Maunz *et al.*, Benchmarking a trapped-ion quantum computer with 30 qubits, *Quantum* **8**, 1516 (2024).
- [9] N. Kanazawa, D. J. Egger, Y. Ben-Haim, H. Zhang, W. E. Shanks, G. Aleksandrowicz, and C. J. Wood, Qiskit experiments: A python package to characterize and calibrate quantum computers, *J. Open Source Software* **8**, 5329 (2023).
- [10] A. Pasquale, S. Efthymiou, S. Ramos-Calderer, J. Wilkens, I. Roth, and S. Carrazza, Towards an open-source framework to perform quantum calibration and characterization, *arXiv:2303.10397*.
- [11] N. Wyderka, A. Ketterer, S. Imai, J. L. Bönsel, D. E. Jones, B. T. Kirby, X.-D. Yu, and O. Gühne, Complete characterization of quantum correlations by randomized measurements, *Phys. Rev. Lett.* **131**, 090201 (2023).
- [12] M. L. Dahlhauser and T. S. Humble, Benchmarking characterization methods for noisy quantum circuits, *Phys. Rev. A* **109**, 042620 (2024).
- [13] M. Q. Khan, W. Dong, L. M. Norris, and L. Viola, Multiaxis quantum noise spectroscopy robust to errors in state preparation and measurement, *Phys. Rev. Appl.* **22**, 024074 (2024).
- [14] M. Urbanek, B. Nachman, V. R. Pascuzzi, A. He, C. W. Bauer, and W. A. de Jong, Mitigating depolarizing noise on quantum computers with noise-estimation circuits, *Phys. Rev. Lett.* **127**, 270502 (2021).
- [15] P. Zhao, K. Linghu, Z. Li, P. Xu, R. Wang, G. Xue, Y. Jin, and H. Yu, Quantum crosstalk analysis for simultaneous gate operations on superconducting qubits, *PRX Quantum* **3**, 020301 (2022).
- [16] T. Proctor, M. Revelle, E. Nielsen, K. Rudinger, D. Lobser, P. Maunz, R. Blume-Kohout, and K. Young, Detecting and tracking drift in quantum information processors, *Nat. Commun.* **11**, 5396 (2020).
- [17] J. Wang, S. Paesani, R. Santagati, S. Knauer, A. A. Gentile, N. Wiebe, M. Petruzzella, J. L. O'Brien, J. G. Rarity, A. Laing *et al.*, Experimental quantum Hamiltonian learning, *Nat. Phys.* **13**, 551 (2017).
- [18] Z. An, J. Wu, M. Yang, D. Zhou, and B. Zeng, Unified quantum state tomography and Hamiltonian learning: A language-translation-like approach for quantum systems, *Phys. Rev. Appl.* **21**, 014037 (2024).
- [19] W. Yu, J. Sun, Z. Han, and X. Yuan, Robust and efficient Hamiltonian learning, *Quantum* **7**, 1045 (2023).
- [20] A. Gu, L. Cincio, and P. J. Coles, Practical Hamiltonian learning with unitary dynamics and Gibbs states, *Nat. Commun.* **15**, 312 (2024).
- [21] A. Dutkiewicz, T. E. O'Brien, and T. Schuster, The advantage of quantum control in many-body Hamiltonian learning, *Quantum* **8**, 1537 (2024).
- [22] M. Mohseni, A. T. Rezakhani, and D. A. Lidar, Quantum-process tomography: Resource analysis of different strategies, *Phys. Rev. A* **77**, 032322 (2008).
- [23] E. Flurin, L. S. Martin, S. Hacoen-Gourgy, and I. Siddiqi, Using a recurrent neural network to reconstruct quantum dynamics of a superconducting qubit from physical observations, *Phys. Rev. X* **10**, 011006 (2020).
- [24] K. Tucker, A. K. Rege, C. Smith, C. Monteleoni, and T. Albash, Hamiltonian learning using machine-learning models trained with continuous measurements, *Phys. Rev. Appl.* **22**, 044080 (2024).
- [25] T. J. Evans, R. Harper, and S. T. Flammia, Scalable Bayesian Hamiltonian learning, *arXiv:1912.07636*.
- [26] A. Dutt, E. Pednault, C. W. Wu, S. Sheldon, J. Smolin, L. Bishop, and I. L. Chuang, Active learning of quantum system Hamiltonians yields query advantage, *Phys. Rev. Res.* **5**, 033060 (2023).
- [27] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* **378**, 686 (2019).
- [28] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, Physics-informed neural networks (PINNs) for fluid mechanics: A review, *Acta Mech. Sin.* **37**, 1727 (2021).
- [29] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, Scientific machine learning through physics-informed neural networks: Where we are and what's next, *J. Sci. Comput.* **92**, 88 (2022).
- [30] L. K. Castelano, I. Cunha, F. S. Luiz, R. de Jesus Napolitano, M. V. d. S. Prado, and F. F. Fanchini, Combining physics-informed neural networks with the freezing mechanism for general Hamiltonian learning, *Phys. Rev. A* **110**, 032607 (2024).
- [31] F. Davoodi, Active physics-informed deep learning: Surrogate modeling for nonplanar wavefront excitation of topological nanophotonic devices, *Nano Lett.* **25**, 768 (2025).
- [32] M. Zhu, H. Zhang, A. Jiao, G. E. Karniadakis, and L. Lu, Reliable extrapolation of deep neural operators informed by physics or sparse observations, *Comput. Methods Appl. Mech. Eng.* **412**, 116064 (2023).
- [33] X. Yang, S. Zafar, J.-X. Wang, and H. Xiao, Predictive large-eddy-simulation wall modeling via physics-informed neural networks, *Phys. Rev. Fluids* **4**, 034602 (2019).
- [34] W. Zhou and Y. Xu, Data-guided physics-informed neural networks for solving inverse problems in partial differential equations, *arXiv:2407.10836*.
- [35] Y. Chen, L. Lu, G. E. Karniadakis, and L. Dal Negro, Physics-informed neural networks for inverse problems in nano-optics and metamaterials, *Opt. Express* **28**, 11618 (2020).

- [36] P. Jin, Z. Zhang, A. Zhu, Y. Tang, and G. E. Karniadakis, SympNets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems, *Neural Networks* **132**, 166 (2020).
- [37] S. Greydanus, M. Dzamba, and J. Yosinski, Hamiltonian neural networks, *Adv. Neural Inf. Process. Syst.* **32**, 15353 (2019).
- [38] N. Wittler, F. Roy, K. Pack, M. Werninghaus, A. S. Roy, D. J. Egger, S. Filipp, F. K. Wilhelm, and S. Machnes, Integrated tool set for control, calibration, and characterization of quantum devices applied to superconducting qubits, *Phys. Rev. Appl.* **15**, 034080 (2021).
- [39] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Neural-network quantum state tomography, *Nat. Phys.* **14**, 447 (2018).
- [40] Z.-A. Jia, B. Yi, R. Zhai, Y.-C. Wu, G.-C. Guo, and G.-P. Guo, Quantum neural network states: A brief review of methods and applications, *Adv. Quantum Technol.* **2**, 1800077 (2019).
- [41] I. L. Gutiérrez and C. B. Mendl, Real time evolution with neural-network quantum states, *Quantum* **6**, 627 (2022).
- [42] V. Bauduin, S. Cuomo, and V. S. Di Cola, Impact of collocation point sampling techniques on pinn performance in groundwater flow predictions, *J. Comput. Math. Data Sci.* **14**, 100107 (2025).
- [43] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, Learning nonlinear operators via deeponet based on the universal approximation theorem of operators, *Nat. Mach. Intell.* **3**, 218 (2021).
- [44] R. Wang, P. Zhao, Y. Jin, and H. Yu, Control and mitigation of microwave crosstalk effect with superconducting qubits, *Appl. Phys. Lett.* **121**, 152602 (2022).
- [45] J. B. Hertzberg, E. J. Zhang, S. Rosenblatt, E. Magesan, J. A. Smolin, J.-B. Yau, V. P. Adiga, M. Sandberg, M. Brink, J. M. Chow *et al.*, Laser-annealing josephson junctions for yielding scaled-up superconducting quantum processors, *npj Quantum Inf.* **7**, 129 (2021).
- [46] Z. H. Yang, R. Wang, Z. T. Wang, P. Zhao, K. Huang, K. Xu, Y. Tian, H. F. Yu, and S. P. Zhao, Mitigation of microwave crosstalk with parameterized single-qubit gate in superconducting quantum circuits, *Appl. Phys. Lett.* **124**, 214001 (2024).
- [47] O. Marschall, K. Cho, and C. Savin, A unified framework of on-line learning algorithms for training recurrent neural networks, *J. Mach. Learn. Res.* **21**, 1 (2020).
- [48] J. Liu and X. Wang, Code and data for the manuscript “Hamiltonian learning via inverse physics-informed neural networks”, GitHub, <https://github.com/jliu-boat/ipinnhl> 2025, <https://github.com/jliu-boat/ipinnhl>.
- [49] G. E. Hinton, A practical guide to training restricted boltzmann machines, in *Neural Networks: Tricks of the Trade*, edited by G. Montavon, G. B. Orr, and K.-R. Müller (Springer, Berlin, 2012), 2nd ed., pp. 599–619.
- [50] D. Hendrycks and K. Gimpel, Gaussian error linear units (GELUs), [arXiv:1606.08415](https://arxiv.org/abs/1606.08415).