New Journal of Physics
The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG
IOP Institute of Physics

Published in partnership with: Deutsche Physikalische Gesellschaft and the Institute of Physics

OPEN ACCESS

RECEIVED
22 December 2021

REVISED
15 April 2022

ACCEPTED FOR PUBLICATION
17 May 2022

PUBLISHED
6 June 2022

PAPER

Neural networks for quantum inverse problems

Ningping Cao[1,2,*], Jie Xie[3,4], Aonan Zhang[3,4], Shi-Yao Hou[5], Lijian Zhang[3,4] and Bei Zeng[6]

[1] Department of Mathematics & Statistics, University of Guelph, Guelph N1G 2W1, Ontario, Canada
[2] Institute for Quantum Computing, University of Waterloo, Waterloo N2L 3G1, Ontario, Canada
[3] National Laboratory of Solid State Microstructures, College of Engineering and Applied Sciences and School of Physics, Nanjing University, Nanjing 210093, People's Republic of China
[4] Collaborative Innovation Center of Advanced Microstructures, Nanjing University, Nanjing 210093, People's Republic of China
[5] College of Physics and Electronic Engineering, Center for Computational Sciences, Sichuan Normal University, Chengdu 610068, People's Republic of China
[6] Department of Physics, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, People's Republic of China
[*] Author to whom any correspondence should be addressed.

E-mail: ncao@uwaterloo.ca, lijian.zhang@nju.edu.cn and zengb@ust.hk

Keywords: quantum information, quantum machine learning, quantum inverse problem

Abstract

Quantum inverse problem (QIP) is the problem of estimating an unknown quantum system from a set of measurements, whereas the classical counterpart is the inverse problem of estimating a distribution from a set of observations. In this paper, we present a neural-network-based method for QIPs, which has been widely explored for its classical counterpart. The proposed method utilizes the quantumness of the QIPs and takes advantage of the computational power of neural networks to achieve remarkable efficiency for the quantum state estimation. We test the method on the problem of maximum entropy estimation of an unknown state ρ from partial information both numerically and experimentally. Our method yields high fidelity, efficiency and robustness for both numerical experiments and quantum optical experiments.

1. Introduction

Learning quantum states is an essential task in quantum information processing [1–3]. Typically, performing measurements on a quantum system, getting readouts, and reconstructing the quantum states is the way to study the corresponding systems. In general, this process can be written as $\vec{c} = \mathcal{A}(\rho) + \vec{\epsilon}$, where ρ is the quantum state of the system, $\mathcal{A}$ is a function of ρ, and $\vec{c}$ is expectation values obtained from the measurements that are specified by the function $\mathcal{A}$. The vetcor $\vec{\epsilon}$ is the noise vector that is subject to a noise distribution π_{noise}. If one knows the state ρ, then $\vec{c}$ can be obtained by getting the measurements specified by $\mathcal{A}$, and we call this process the quantum forward problem (QFP); the opposite direction is then called the quantum inverse problem (QIP).

Despite of the contexts of quantum systems, when the operator ρ is a diagonal matrix, i.e. a classical probability distribution, the QIP reduces to its classical counterpart, i.e. the classical inverse problem (IP), which is known as one of the most important mathematical problems since it estimates parameters that are not directly measurable, a situation widely applicable in science and technology. From the analytic perspective, it is a problem of recovering system parameters (elements of diagonal ρ) from observables $\vec{c}$ according to a particular model; from the Bayesian inference viewpoint, the goal is to recover the classical probability distribution of a diagonal ρ subject to the given $\vec{c}$ [4].

IP has been studied for a long time in statistical inference and statistical learning [5, 6]. One of the difficulties is that IPs are often *ill-posed*. That means the solution may not exist, not unique when it exists, or unstable in the sense that a slight change in the input may cause a significant change to the output. Deep neural networks are state-of-art tools for solving IP [4, 7–12], with several of them focused on the ill-posed

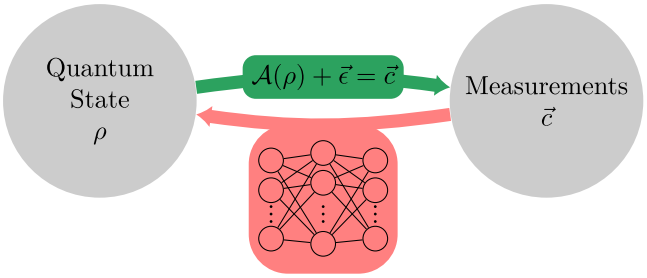


Figure 1. The schematic diagram of the general method: the green arrow depicts the quantum forward model; the red arrow is the QIP. We tackle the problem by supervised learning a deep neural network.

cases [4, 7]. These proposals have been broadly applied to biomedical imaging [8, 13–16], geophysics [17], optical physics [18], and engineering sciences [19] etc.

When $\mathcal{A}$ is the function of directly mapping ρ to complete measurements $\vec{c}$, the QIP corresponds to the so-called quantum state tomography. The standard tomography process requires measurements from $d^2 - 1$ linearly independent observables, where d is the dimension of Hilbert space [20]. It is highly resource-demanding since the number of measurements grows exponentially with the number of qubits, making it impractical even in the noisy intermediate-scale quantum (NISQ) era. In general, the goal of QIPs, from a statistical inference perspective, is to discover the quantum distributions based on observed $\vec{c}$ and then give an estimation based on the distribution. Quantum probability (non-diagonal ρ) is a noncommutative probability on von Neumann algebra, while classical probability is the commutative special case (its von Neumann algebra is abelian) [21]. This difference makes quantum probability distributions even harder to learn. Deep neural networks have great potential to give sound estimations in such challenging scenarios. In this paper, we develop a method for QIPs using deep neural networks, which is shown in figure 1. We remark that simple networks are not the only tools for QIPs. There are series of techniques for classical IP that suit for particular contexts, for example, other network structures [22, 23], specially designed loss function [4], all sorts of regularizer [7, 24] etc, one can conveniently modify them for QIPs. To demonstrate our idea clearly and without loss of generality, we only present it in the language of simple deep networks.

After properly parametrizing the quantum state ρ, deep neural networks offer an excellent opportunity to fulfill this task by implementing supervised learning techniques. The common concerns of applying supervised learning techniques are the availability of training data and distribution. In our case, the knowledge from quantum physics pitches in, providing ways of parameterizing states and physical models for the QFP. It can generate abundant training and testing data; it also reveals the landscape of the QIP, which can determine the training data sampling and distribution.

We consider incomplete measurements to demonstrate our deep learning method since the complete measurements require $d^2 - 1$ operators, which are resource-demanding. An incomplete set of measurements $\vec{c}$, reference to as partial information, usually does not reveal all system information. A given $\vec{c}$ usually corresponds to multiple preimages ρ. However, with appropriate prior information, the partial information could have an almost (with probability one) bijective relation with the quantum system. For example, based on the knowledge that the unknown quantum state has a low rank, compressed sensing [25–27] can reconstruct the state with Pauli operators much less than $d^2 - 1$; a small number of measurements can predict many properties in a quantum system [28, 29]; with certain assumptions, one eigenstate can encode all the information in a Hamiltonian [30–32]. The latter means the measurements of one eigenstate can help recover the system Hamiltonian and the measured eigenstate. Though the mathematical connection $\mathcal{A}$ between ρ and expectation values $\vec{c}$ is clear, it is nontrivial to come up with effective schemes to solve the QIP while efficiently utilizing the prior information [26].

Our method is particularly effective for these problems with prior information. The almost bijective relation guaranteed by prior information of the system restricts the corresponding QIP to be likely well-posed. The method can easily and effectively utilize the prior information of a regarding quantum system because the information can easily embed in the QFP. The design of specific steps for a giving QIP is intuitive, as well as the network training. Our framework can not only be applied to solve the problem but also offer initial values for other Monte Carlo-based methods. We test our scheme on the task of giving maximum entropy estimation (MEE) from partial information. The method demonstrates high fidelity and extraordinary efficiency on both numerical data and experimental data from optical devices. It also manifests the ability to tolerate experimental noise.

This paper is organized in the following way: section 2 presents and discusses the deep learning method; section 3 demonstrates the example of giving the MEE ρ_{MEE} of an unknown state ρ from partial information in detail, supported by both numerical and quantum optical experimental results; section 4 then follows with discussion and outlook.

2. The neural networks method

For a quantum physical system, the underlying mechanism is governing by the physical model $\mathcal{A}_{\mathbf{F}} : X \to Y$ such that

$$\vec{c} = \mathcal{A}_{\mathbf{F}}(\rho) + \vec{\epsilon}, \tag{1}$$

where $\rho \in X$ is the density matrix, $\vec{\epsilon} \in Y$ is the noise vector (subject to a noise distribution π_{noise}), and $\vec{c} \in Y$ is the expectation values of measuring a fixed set of observables $\mathbf{F} = \{F_1, \ldots, F_m\}$. X is the interested set of $d \times d$ density matrices according to given prior information. For example, if we know the states ρ are pure states, X is then the set of rank-1 density matrices; X is the set of all density matrices if no prior information is provided. Y is the vector space $\mathbb{R}^m$. The physical model $\mathcal{A}_{\mathbf{F}}$ is always associated with a set $\mathbf{F}$ of observables to determine which space it maps into. For simplicity, we will use the notation $\mathcal{A}$ instead. Giving ρ to get $\vec{c}$ is the QFP, the reverse direction we call the QIP.

When the measurements are not complete—that is, the number m of observables is not enough to uniquely determine the system, the QIP does not possess a unique solution. Prior information about the interested system decreases the degree of freedom. Under this prior information, a model $\mathcal{A}$ is indeed bijective or almost bijective ('almost' in this context means 'measure one', i.e. the function is bijection except for a measure zero set of exceptions.) That means the system can be reconstructed with fewer measurements if the prior information has been appropriately used. However, the challenge of how to efficiently encode the prior information into a practical scheme is demanding [26]. This turns out to be a blessing for our framework, which is straightforward to utilize such information. This point will be explained later.

The key observation is that the forward problem is almost always more straightforward than the inverse. For example, reconstructing an object from its projections is substantially more challenging than making the projections of a known object. Especially in quantum information theory, the forward problem is usually clear thanks to the study of quantum physics. Knowing the information of a quantum system, such as its Hamiltonian or state (density matrix), the measurement outcomes of this system according to a fixed set $\mathbf{F}$ are predictable. Compared to reconstructing information about the system from its measurement outcomes, the forward direction is significantly easier.

We take advantage of the complexity difference between the two directions and use the easier direction to help deal with the problematic side. Supervised learning is the relatively mature branch of machine learning techniques that finds the model between input and output pairs data with labelled examples. On the contrary, unsupervised learning techniques are normally used while lack access to training data or having trouble with labelling data [33]. In our problem, data resource for supervised learning is guaranteed. The QFP contributes to generating training and testing data for supervised learning. The next problem is the training data distribution. The forward model $\mathcal{A}$ contains the information about the landscape of $\vec{c}$ according to ρ. This information guides the training data sampling process, largely determining the training data distribution.

From another perspective, QIPs are also regression problems to fit given data pairs. Neural networks are incredibly versatile tools for regression problems. Even simple NNs only with one hidden layer are very expressive. With nonlinear activation functions between neurons, these 'vanilla' NNs can represent arbitrary functions [34]. Traditionally, regression problems require deliberately chosen techniques to achieve better performance. However, NNs are extremely flexible. They automatically adapt themselves to different regression techniques according to the particular scenario. This feature made NN a convenient tool for solving various QIPs.

Before implementation, we need to parametrize density matrices ρ. The parametrization function $P : X' \to X$ is a bijection,

$$P(\vec{a}) = \rho, \tag{2}$$

where $\vec{a} \in X'$, X' is a vector space. The choice of P is based on X. For example, if the set X is the Gibbs states of a class of Hamiltonian, P could be the map from Hamiltonian parameters to the Gibbs states.

The training data set is denoted as

$$\mathbf{T} = \{(\vec{c}_{\,\mathrm{train}}, \vec{a}_{\,\mathrm{train}}) | \vec{c}_{\,\mathrm{train}} = (\mathcal{A} \circ P)(\vec{a}_{\,\mathrm{train}}) + \vec{\epsilon}, \vec{a}_{\,\mathrm{train}} \in X'_{\mathrm{train}}\},$$

where $X'_{\text{train}} \subset X'$ is the finite set of sampled parameters $\vec{a}_{\text{train}}$. The training data naturally implants the prior information contained in $\mathcal{A}$.

After data preparation, the network can be trained. A neural network is a tunable function $\Phi_{\text{NN}} : Y \to X'$. The training process uses training algorithms to tune the parameters embedded in Φ_{NN} to minimize the distances between the NN output and desired output according to the chosen loss function, i.e. minimizing

$$L(\vec{a}_{\text{train}}, \Phi_{\text{NN}}(\vec{c}_{\text{train}})), \tag{3}$$

where $L : X' \times X' \to \mathbb{R}$ is the loss function. It is chosen to reverberate the parametrization P. The goal is to bring ρ_1 and ρ_2 closer by minimizing $L(\vec{a}_1, \vec{a}_2)$. The loss function L can be the mean square error or mean absolute error (MAE) if $\vec{a}$s need to be precisely approached on magnitudes. If P focus more on the direction of $\vec{a}$s, a loss function that minimizes angle (e.g. cosine similarity) will be a better option. L can also be a type of entropy when $\vec{a}$s are probability distributions. The choice of P and L, as well as the training data set $\mathbf{T}$ all reflect prior information of the problem.

For testing data generated by the QFP, comparing the ideal ρ and the estimated ρ_{est} can tell us the accuracy of the estimation. A reasonable question to ask is, given a data $\vec{c}_{\text{unk}}$ with an unknown preimage, how we can know whether the NN estimation ρ_{est} is acceptable. It turns out that QFP can serve as the mechanism of examining the estimation that come out from a trained NN. Choosing a metric f in Y, one can compare $\vec{c}_{\text{unk}}$ and the image of NN output,

$$f(\vec{c}_{\text{unk}}, \mathcal{A} \circ P \circ \Phi_{\text{NN}}(\vec{c}_{\text{unk}})). \tag{4}$$

Ideally, we want $\vec{c}_{\text{unk}}$ and $(\mathcal{A} \circ P \circ \Phi_{\text{NN}})(\vec{c}_{\text{unk}})$ to be identical, but numerical errors are inevitable in reality. Bounding the value of equation (4) can bound the confidence of ρ_{est}.

In the next section, we will provide an example to demonstrate our method. The task is to give a MEE based on noiseless partial information of an unknown state. The network takes incomplete measurements of the unknown quantum state ρ and returns the MEE of ρ. Compared to other algorithms, our method shows extraordinary efficiency without sacrificing much accuracy. It also shows a remarkable ability to tolerate experimental error.

3. Learning maximum entropy estimation from partial information

Maximum entropy inference is believed to be the best estimation to present the current knowledge when only part of the information about the system is provided [35, 36]. The entropy is mostly Shanon entropy in classical physics and engineering, and is von Neumann entropy for the quantum counterpart.

In quantum system, given the set of incomplete measurement expectation values $\{c_i | c_i = \text{tr}(\rho F_i), F_i \in \mathbf{F}\}$ of an unknown state ρ for a fixed set of observables $\mathbf{F}$, there may exist more than one quantum state with the same measurement outcomes. The incompleteness means that the measurements are insufficient for a full tomography ($m < d^2 - 1$). Denote the set of states as

$$\mathbf{P} = \{\rho^* | \text{tr}(\rho^* F_i) = \text{tr}(\rho F_i), \forall\, F_i \in \mathbf{F}\}.$$

The unknown state ρ is one of the elements in $\mathbf{P}$. The MEE ρ_{MEE} of ρ can be represented as a thermal state

$$\rho_{\text{MEE}} = \frac{\exp\left(\beta \sum_i a_i F_i\right)}{\text{tr}\left[\exp\left(\beta \sum_i a_i F_i\right)\right]}, \tag{5}$$

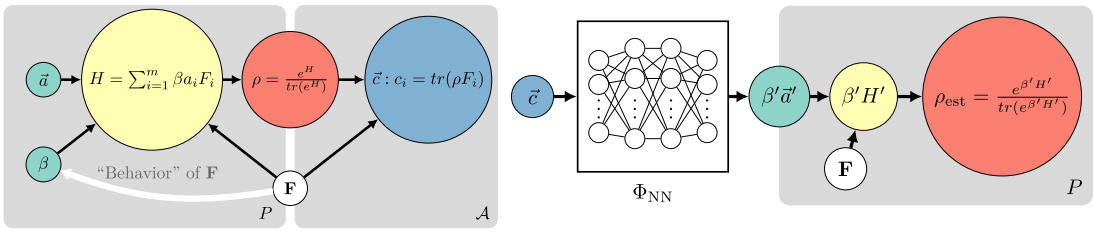
where β is the reciprocal temperature of the system and a_i's are real coefficients [37–40]. At the mean time, ρ_{MEE} should satisfy that it has the same measurement outcomes when measuring the same set of operator $\mathbf{F}$ (i.e. $\rho_{\text{MEE}} \in \mathbf{P}$). The thermal representation is unique [37]. The measurement results $\vec{c} = (c_1, \ldots, c_m)$ where $c_i = \text{tr}(\rho F_i), F_i \in \mathbf{F}$, therefore, possess a one-to-one correspondence with its MEE ρ_{MEE}.

An interesting special case is that $\mathbf{P}$ only has one element, then $\rho = \rho_{\text{MEE}}$. A well-studied example of such case is when ρ is an unique ground state of $H = -\sum_i a_i F_i$, where $F_i \in \mathbf{F}$ and $a_i \in \mathbb{R}$ [41–43]. This situation means that if ρ is a unique ground state of H, ρ_{MEE} not only has a one-to-one correspondence with $\vec{c}$, but also is the actual state ρ.

In this particular QIP, the parametrization function P and the forward model $\mathcal{A}$ are as follows:

$$P : \beta\vec{a} \to \rho_{\text{MEE}} \tag{6}$$

$$\mathcal{A} : \rho_{\text{MEE}} \to \vec{c}$$



(a) Training and testing data generation (b) The process

Figure 2. (a) β and $\vec{a}$ are randomly generated. The Hermitian operator set **F** associated with the forward model $\mathcal{A}$ provides information for the distribution of β. Generated pairs of $(\vec{c}, \beta\vec{a})$ are training data: $\vec{c}$ is the input of the neural network, $\beta\vec{a}$ is the ideal output. The left block is the parametrization function P, and the right block is the foreword model $\mathcal{A}$ for this problem. (b) The trained network Φ_{NN} is the approximation of $P^{-1} \circ \mathcal{A}^{-1}$, which can produce a parameter estimation $\beta'\vec{a}'$ according to an input $\vec{c}$. The right block is the parameterization function P that maps estimated parameters $\beta'\vec{a}'$ to ρ_{est}.

where $\vec{a} = (a_1, \ldots, a_m)$ and ρ_{MEE} is defined in equation (5). The noise is set to be zero, i.e. $\vec{\epsilon} = \vec{0}$. Supervised learning can train a network Φ_{NN} to approach the inverse function

$$P^{-1} \circ \mathcal{A}^{-1} : \vec{c} \to \beta\vec{a}. \tag{7}$$

More specifically, as shown in figure 2(a), we randomly generate many β's and $\vec{a}$'s, achieving corresponding measurement results $\vec{c}$. These pairs of $(\vec{c}, \beta\vec{a})$ are used as training data for the neural network. The trained network Φ_{NN} is the approximation of the function $P^{-1} \circ \mathcal{A}^{-1}$. The estimation of MEE

$$\rho_{est} = \frac{\exp\left(\sum_i \beta' a_i' F_i\right)}{\mathrm{tr}\left[\exp\left(\sum_i \beta' a_i' F_i\right)\right]}$$

from $P \circ \Phi_{NN}(\vec{c})$, where β' and a_i' are NN estimations of the true values β and a_i. To be noticed that when the unknown state ρ is not a thermal state, the operator $-H' = -\sum_i a_i' F_i$ is not necessarily the real Hamiltonian of the system. We call H' a *pseudo Hamiltonian*.

We test our method numerically with two systems: (1) **F** has three 64 by 64 random generated Hermitian operators; (2) the five-qubit one-dimension lattice, $\mathbf{F} = \{\sigma_a^{(i)} \otimes \sigma_b^{(i+1)} | \sigma_a, \sigma_b \in \mathcal{P}, 1 \leqslant i \leqslant 4, a + b \neq 0\}$ where $\mathcal{P} = \{\sigma_0 = I, \sigma_1 = \sigma_x, \sigma_2 = \sigma_y, \sigma_3 = \sigma_z\}$ is the set of Pauli operators with the 2 by 2 identity. The upper index i indicates the qubit which the operator acts on. Moreover, we test the method with experimental data of an optical set-up, which are derived from unique ground states associated with fixed Hermitian operator sets. Therefore the MEE estimations ρ_{est} are also the estimation of the true states measured in our experiments.

3.1. Data preparation and network training

Training data preparation is the key to supervised learning since the learning outcome depends heavily on the training data set.

The data generating procedure is shown in figure 2(a). Parameter $\vec{a}$ is drawn from normal $\mathcal{N}(0, 1)$ distribution then normalized. The 'coldness' β is randomly sampled from $(0, 100]$. Generally, when β reaches ~30, thermal states are almost pure. Here we allow β goes to 100 for some extreme cases. The distribution of β in the whole training data set is critical in this process. We will discuss this issue in depth later. Parameter $\vec{a}$ together with the fixed set of operators **F** set up the operator $H = \sum_i \beta a_i F_i$. The measurement results $\vec{c} = (\mathrm{tr}(\rho F_1), \ldots, \mathrm{tr}(\rho F_m))$ come from trace the product of $\rho = \exp(H)/\mathrm{tr}[\exp(H)]$ and operator F_i's. Every pair of $\beta\vec{a}$ and $\vec{c}$ counts for a pair of training data.

It turns out that the distribution of β in the training data set is the key to our problem. By data distribution of β, we mean the proportion of β picked in a given interval I to the amount of data in the whole training data set. Intuitively, the network should be trained with more data in the place where the function changes more rapidly. Specifically, the network should see more data on where the slight change of β causes a significant change on ρ then on $\vec{c}$ in the relative sense. Despite the matrix exponential function, the property also depends on **F**. Luckily enough, since we know **F**, we have all the information we need. The function is steeper while β is small and is smooth while β is relatively large.

However, if we put significantly more data on the narrow steep region (e.g. $\beta \in (0, 5)$), that may confuse the network—the network will have bad performance on the wider smooth region since it does not see enough data. In order to achieve optimal overall performance, one needs to balance between fitting the rough region and giving enough data of other regions.

First of all, we need a way to measure the 'roughness' of the function in a given area according to the parameter β. We choose how far away the thermal state $\rho = \exp(\sum_i \beta a_i F_i)/\mathrm{tr}[\exp(\sum_i \beta a_i F_i)]$ is from being a pure state as the indicator (denote as λ). In other words,

$$\lambda = 1 - \lambda^0,$$

where λ^0 is the biggest eigenvalue of ρ.

We divide β into multiple intervals $I_i = (i, i+1]$ where $0 \leqslant i \leqslant 99$ and $i \in \mathbb{Z}$. In each interval I_i, 1000 data points have been sampled. 1000 βs are drawn from uniform distribution in I_i while 1000 normalized $\vec{a}_i$ is sampled from normal distribution. These βs, $\vec{a}$s and $\mathbf{F}$ together form 1000 $\rho = \exp(\sum_i \beta a_i F_i)/\mathrm{tr}[\exp(\sum_i \beta a_i F_i)]$. Getting λ from each ρ, we calculate the average of these λs and denote it as $\bar{\lambda}_i$. The vector $\vec{\bar{\lambda}} = (\bar{\lambda}_1, \ldots, \bar{\lambda}_N)$ for all intervals is a characterization of the model according to the change of β (denote N as the number of intervals for generality). Let $p_i = \bar{\lambda}_i / \sum_i \bar{\lambda}_i$ and

$$\vec{p}_\beta = (p_1, \ldots, p_N).$$

One may consider using $\vec{p}_\beta$ to be the data distribution. However, it transpires that $\vec{p}_\beta$ is not appropriate since it will concentrate the training data in the lower region.

Referring to our previous arguments, we need to balance the distribution. We take two flattened steps:

(a) Take the average of the first 10 elements in $\vec{\bar{\lambda}}$ and call it $\bar{\lambda}_{\mathrm{ten}}$, then replace these first ten elements which are smaller than $\bar{\lambda}_{\mathrm{ten}}$ with it;

(b) Denote $\sum_i \bar{\lambda}_i / N$ as $\bar{\lambda}_{\mathrm{avrg}}$ and then replace elements which are smaller than $\bar{\lambda}_{\mathrm{avrg}}$ with it.

We normalize the resulting vector and denote it as $\vec{p}_{\mathrm{flat}}$. It is the data distribution we use in this work. Three different data generating methods have been compared in detail in appendix A.

The neural networks used in this work are fully-connected feed-forward. It means that the neurons in one layer are fully connected to the neurons in the next layer, and the information is only passing forward. The input and output layers are determined by the given length of the measurement results (i.e. the cardinality of the fixed operator set $\mathbf{F}$). The three random 64 by 64 operator cases have three input and three output neurons (we refer to it as case 1 later in this paper). The five-qubit 1D lattice example has 51 neurons ($5\binom{3}{1} + 4\binom{3}{1}\binom{3}{1} = 51$) for input and output layers (we call it case 2). These two networks all have two hidden layers; each layer has 100 neurons.

The networks in this work are trained with Adam optimizer [44] which is a popular adaptive learning rate optimization algorithm designed for deep networks. The loss function we chose is MAE

$$L(\vec{e}) = \frac{\sum_i^m |e_i|}{m}$$

where $\vec{e} = \vec{a} - \vec{a}'$ is the error vector between the true value $\vec{a}$ and the estimated value $\vec{a}'$. MAE performs better than mean squared error (MSE, $L(\vec{e}) = \sum_i^m e_i^2/m$, another commonly used loss function). It is because the parametrization function P (equation (6)) would require $\vec{a}'$ to be as close to $\vec{a}$ as possible to bring the images close and the square in MSE will make small errors more indistinct.

For case 1, the number of training data pairs is 3010 470. The batch size is 40 000, and we train the network for 300 epochs. And we use 2005 584 pairs of training data for the five-qubit 1D lattice model. The batch size is 20 000, and the number of epochs is also 300. How the amount of training data affects the accuracy of NNs is analyzed in appendix B.

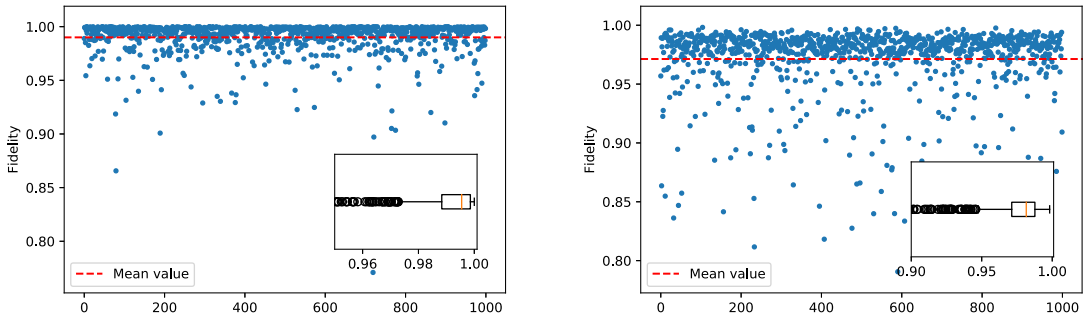
3.2. Numerical results

New data sets are generated to test the performance of trained neural networks. Similar to the procedure of producing training data in figure 2(a), the testing data are pairs of $\vec{c}$ and $\beta\vec{a}$. β's are uniformly picked from $(0, 100]$ and $\vec{a}$'s are normalized.

The estimated MEE ρ_{est} comes out from adopting the course in figure 2(b). We compare each ρ_{est} with its true MEE ρ_{MEE} by calculating the fidelity. The fidelity function we use is given as [20]

$$f(\rho_1, \rho_2) = \mathrm{tr}\left(\sqrt{\sqrt{\rho_1}\rho_2\sqrt{\rho_1}}\right).$$

For case 1, the average fidelity between true MEE ρ_{MEE} and the estimated MEE ρ_{est} is 99.0%. Figure 3(a) shows the fidelities of all tested data. The mini-figure is its boxplot [45], which is a graphical way to depict data through their quartiles. The orange line in the boxplot is the median value which is 99.5%. Statistically,



(a) Case 1: randomly generated operator set $\mathbf{F}$, 3 operators, dimension $d = 64$

(b) Case 2: 5-qubit 1D lattice

Figure 3. Testing results: 1000 pairs of new data are tested for both cases. The mini-plots are the boxplots of these fidelities. The average fidelity for case 1 is 99.0%, for case 2 is 97.1%.

Table 1. Statistics of numerical results.

	Mean	Median	STD
Case 1	99.0%	99.5%	17.0×10^{-3}
Case 2	97.1%	98.1%	31.1×10^{-3}

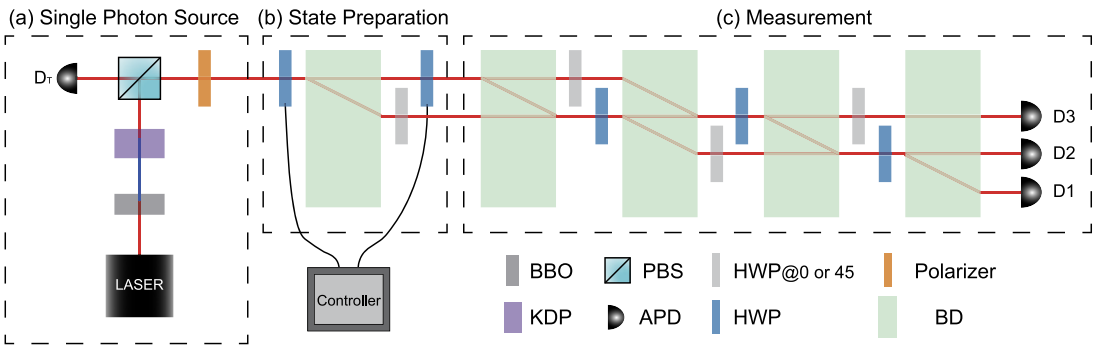


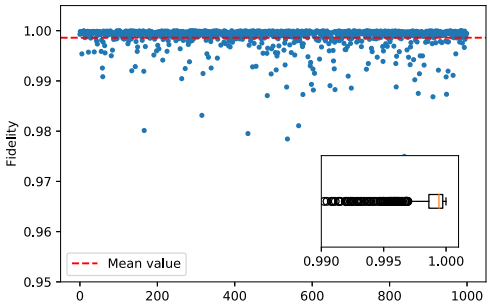
Figure 4. Experimental setup. (a) Single-photon pairs are generated by pumping a phase-matched bulk potassium dihydrogen phosphate crystal with the second harmonics generated from the beta barium borate (β-BBO) crystal. After a polarizing beam splitter, the idler mode is detected by the detector D_T as a trigger of the heralded single-photon source. In contrast, the signal mode is directed towards the following setup. (b) Single photons are prepared as photonic qutrits encoded in the polarization and spatial modes through the HWPs and a calcite beam displacer (BD). (c) The measurement part is composed of wave plates and BDs, forming a three-stage interferometer capable of implementing arbitrary qutrit unitary. After the unitary transformation, photons are detected by three avalanche photodiodes. By setting the wave-plates with different angles, the measurement of different operators can be realized.

the circles in the boxplot are outliers which are data points notably different from others, hence lacking statistical significance. Similarly, figure 3(b) shows the fidelities of the whole testing data set for case 2. The average fidelity is 97.1%, and the median fidelity is 98.1% (table 1).

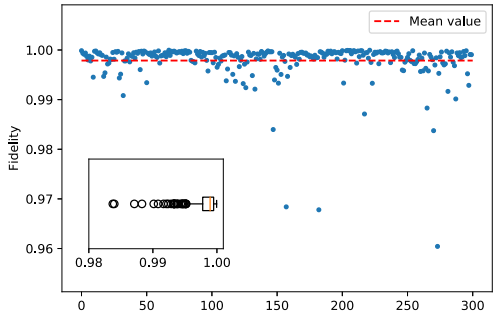
3.3. Experimental verification and the effect of error

To verify the performance of our well-trained neural network in processing actual experimental data and its robustness against experimental noise, we implement a qutrit photonic set-up capable of preparing different qutrit states and measuring arbitrary operators, as shown in figure 4. Particularly, when experimental data are generated by unique ground states of a pseudo-Hamiltonian, ρ_{MEE} is the exact ground state measured, and the NN estimation ρ_{est} is also the approximation of the real state. Therefore, we intentionally prepare ground states of a class of pseudo Hamiltonians and feed them into the measurement devices. By directly comparing the prepared ground states ρ_{exp} with ρ_{est}, we can reveal the network's performance in real experiments.

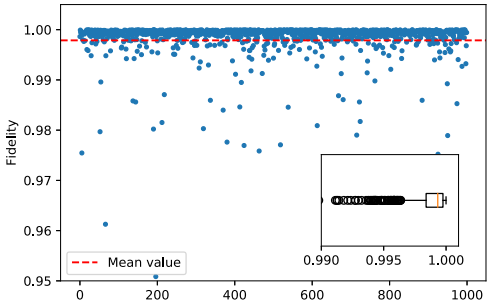
In our experiment, we choose two set of operators $\mathbf{F}_1$ and $\mathbf{F}_2$, and each contain three Hermitian operators. For each set, 300 ground states $\{\rho_{\mathrm{exp}}\}$ of different pseudo Hamiltonian are randomly prepared by changing the setting angles of the two configurable half-wave plates (HWPs) in figure 4(b). Then the



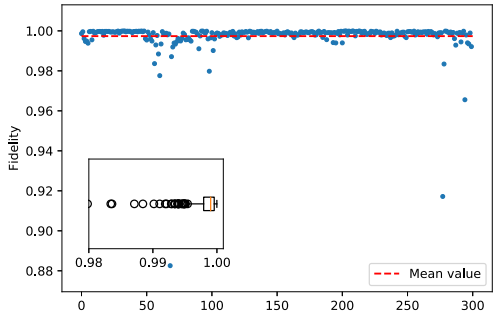
(a) **Numerical testing results of $\mathbf{F}_1$:** Blue dots are the fidelities between the true MEE state ρ_{MEE} and the estimated state ρ_{est}. The x-axis is the dummy variable of the testing set. 1,000 data has been tested, the mean value is 99.9%.



(b) **Fidelities with experimental data of $\mathbf{F}_1$:** The horizontal axis is the dummy label of experimental data points. The average fidelity for all 300 data points is 99.8%. The median value is 99.9% (orange line in the boxplot).



(c) **Numerical testing results of $\mathbf{F}_2$:**Blue dots are the fidelities between the true MEE state ρ_{MEE} and the estimated state ρ_{est}. The x-axis is the dummy variable of the testing set. 1,000 data has been tested, the mean value is 99.8%.



(d) **Fidelities with experimental data of $\mathbf{F}_2$:** The average fidelity of 300 data points is 99.7%. The median value shows in the boxplot is 99.9%.

Figure 5. Numerical and experimental fidelities of $\mathbf{F}_1$ and $\mathbf{F}_2$.

prepared states are input into the measurement part, which is constituted by wave plates, calcite beam displacers (BDs) and photon detectors, capable of projecting the input states into an arbitrary basis. From the measurement statistics, expectation values of different operators can be estimated. Thus by this preparation-and-measurement set-up, we obtain the experimental data set $\vec{c}_{\mathrm{exp}} = (c_{1,\mathrm{exp}}, c_{2,\mathrm{exp}}, c_{3,\mathrm{exp}})$. See appendix C for details.

Before feeding experimental data into the neural networks, we have trained the networks individually for each operator set. 1010 196 and 1003 808 pairs of noiseless (meaning $\vec{\epsilon} = \vec{0}$) numerical data have been used to train networks for $\mathbf{F}_1$ and $\mathbf{F}_2$, respectively. The network structure and other settings (e.g. training algorithm, loss function etc) are in similar fashion with the previous numerical cases. Figure 5(a) shows the numerical results of $\mathbf{F}_1$ for 1000 random generated data. The average fidelity is 99.9%. Figure 5(c) shows the testing fidelities for $\mathbf{F}_2$, and the mean value is 99.8%.

The well-tuned neural networks are now ready for experimental data. Measurement outcomes $\vec{c}_{\mathrm{exp}}$ derived from the experiments are inputs of the networks. From the output parameter set $\beta'\vec{a}'$, the estimated MEEs ρ_{est}s can be derived. The fidelities between ρ_{exp} and ρ_{est} have been calculated and are shown in figure 5(b) ($\mathbf{F}_1$) and figure 5(d) ($\mathbf{F}_2$). The mean value of all 300 data points is 99.8% for $\mathbf{F}_1$, and 99.7% for $\mathbf{F}_2$.

In this experiment, the measurement outcomes suffer from different systematic errors, such as inaccuracies of wave plate setting angles, imperfect interference visibility and drifting of the interferometers, and statistical fluctuations. The average relative errors of different operators $((c_{i,\mathrm{exp}} - \mathrm{tr}(\rho_{\mathrm{exp}}F_i)/\mathrm{tr}(\rho_{\mathrm{exp}}F_i))$ range in $0.79\% \sim 2.43\%$ (see more details in appendix C). Even in this level of additional experimental errors, the networks show almost the same performance in processing the experimental data compared with the numerical data.

3.4. Comparison with other methods

The MEE

$$\rho_{\mathrm{MEE}} = \frac{\exp\left(\beta \sum_i a_i F_i\right)}{\mathrm{tr}\left[\exp\left(\beta \sum_i a_i F_i\right)\right]},$$

for given $\vec{c} = (\mathrm{tr}(\rho F_1), \ldots, \mathrm{tr}(\rho F_m))$ is an optimization problem. It is closely related to the field of information geometry, statistical inference, and machine learning.

An iterative algorithm based on the information-geometry viewpoint is proposed in [37], which runs as follows. First, initialize the system Hamiltonian as an identity operator $H = I$, so the initial density matrix $\rho_{\mathrm{ini}} = \exp(I)/\mathrm{tr}[\exp(I)]$ is the maximum mixed state. The following task is to solve the equations $\mathrm{tr}(\rho F_i) = \mathrm{tr}(\tau F_i)$ for each i, or, to be more precisely, find a density matrix τ to minimize $\sum_i |\mathrm{tr}(\rho F_i) - \mathrm{tr}(\tau F_i)|$. This is done by iteratively update the Hamiltonian H by $H + \epsilon F_i$, so that the density matrix τ is updated as

$$\tau = \frac{\mathrm{e}^H}{\mathrm{tr}(\mathrm{e}^H)} \to \tau' = \frac{e^{H+\epsilon F_i}}{\mathrm{tr}(\mathrm{e}^{H+\epsilon F_i})},$$

in which the parameter ϵ is something like a gradient and can be approximated as

$$\epsilon = \frac{\mathrm{tr}(F_i\rho) - \mathrm{tr}(F_i\tau)}{\mathrm{tr}(F_i^2\tau) - [\mathrm{tr}(F_i\tau)]^2}$$

for each F_i. Repeat the iteration for several rounds, and we can find a τ as closely to ρ as possible.

Another related method is based on the so-called quantum Boltzmann machine (QBM) [46]. The QBM uses a different loss function (or objective function) for optimization, i.e. the cross-entropy,

$$\mathcal{L} = -\sum_i p_i \log p_i',$$

with p_i and p_i' are probability distributions: p_i is the ideal case and p_i' depending on some parameters. The learning process of a QBM is to find certain parameters to minimize $\mathcal{L}$. Take $p_i = C\,\mathrm{tr}(\rho F_i)$ and $p_i' = C'\,\mathrm{tr}(\tau F_i)$, where C and C_i' are normalization constants. The density matrix τ here can also be expressed as $\tau = \exp(H)/\mathrm{tr}[\exp(H)]$. Since $H = \sum_i a_i F_i$, the loss function is now a function of a_is. The loss function $\mathcal{L}$ reaches its minimum for $p_i = p_i'$, so our goal is to optimize $\mathcal{L}$ over possible a_is.

We can use the same method that the QBM uses to learn the maximum entropy state. To use the cross entropy, for F_i with negative eigenvalues, we first renormalize p_i's by adding $(\lfloor -f_{i\,\min} \rfloor + 1)I$ to F_i, where $f_{i\,\min}$ is the lowest eigenvalue of F_i. This ensures p_i's being positive, and adding unity operator to Hamiltonian has no effect on its thermal state. Second, since the p_i and p_i' in cross entropy are probability distributions, which means $\sum_i p_i$ and $\sum_i p_i'$ are both restricted to 1, we add normalization constants C and C' in front of $\mathrm{tr}(\rho F_i)$ and $\mathrm{tr}(\tau F_i)$, respectively.

We test both the iterative algorithm and the QMB algorithm using MATLAB for the examples in appendix B. The iterative algorithm converges to the desired results precisely and effectively. The average time for an iterative algorithm for each case is about 0.0425 s. As a comparison, if we run the optimization using the functions provided by MATLAB, the time for each case is about 0.0148 s.

However, the method of QBM cannot provide a precise approximation to the original density matrix. This fact may be because the gradient is hard to obtain (notice that the forms of the matrix F_i in our cases are far more complicated than the ones discussed in QBM (see [46]). Also, it may be due to the normalization of p_is we have introduced, which can introduce more issues in the learning process. There may be ways to improve the training method, which we will leave for future investigation.

Given that the iterative algorithm seems more effective and accurate for optimization, we compare our supervised learning method with the iterative algorithm. For the case that the measured set $\mathbf{F}$ possesses three 64 by 64 Hermitian operators, our method estimates the test set with 99.0% average fidelity (section 3.2), setting the error bound as 10^{-10}. As a comparison, the iterative algorithm provides the outcome states with the fidelity of almost 1 for every data point. In terms of accuracy, the interactive algorithm is slightly stronger than ours.

By using the same computational device [47], our network can predict 5000 data in less than a second, while the iterative method requires about 10 min for 100 data. In this sense, our method is more efficient for estimation once trained without sacrificing much accuracy.

4. Discussion

This paper presents a deep learning method for QIPs. The method shows good performances for both numerical and experimental data, and robustness against experimental noise. As mentioned in the introduction, other techniques in IP can also be modified to solve particular QIPs. For conciseness and consistency, we only discuss simple networks in this paper. This choice left a vast room for future study.

The example we demonstrated is a quantum state learning problem, with the training data numerically generated. The network can also be trained with noisy experimental data to output idea (noiseless) states.

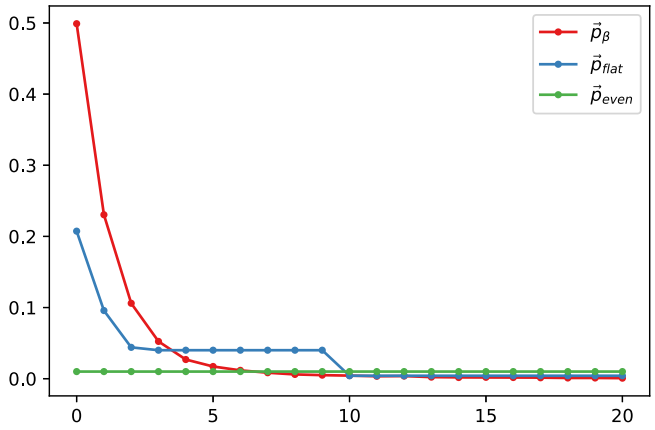


Figure 6. The three distributions for the operator set $\mathbf{F}_1$: the x-axis is the label i of the interval $I_i = (i, i+1]$ for β. The green, blue, red dots depict the even distribution $\vec{p}_{\text{even}}$, the flattened distribution $\vec{p}_{\text{flat}}$ and $\vec{p}_\beta$ respectively.

The outcomes from the trained network would naturally mitigate the experimental error. One can use deliberate approaches to fight against noise in data, such as different NN structures, regularizers, training algorithms, and loss functions [53]. Our method can also be easily adapted to other setups such as Hamiltonian learning. An example and detailed discussion can be found in [48].

We show that our method is straightforward to implement for scenarios that prior information grants a mostly bijective relation between ρ and $\vec{c}$. It can also be used to examine if the prior information is adequate to ensure the map is almost bijective. The 'almost' here means that it is true except a measure zero set of exceptions. After properly implementing this scheme, if the trained network does not perform well, we may consider that the prior information is not exactly pinned down an almost one-to-one correspondence.

For more ill-posed problems, there are several techniques can be applied, such as various regularizers [7], different network structures [14], statistical estimators [11, 49], depending on the particular problem.

Last but not least, the neural networks in the inverse process can be replaced by quantum neural networks [50–52]. In this case, we may not need the parametrization process P. The modified method has the potential to implement on NISQ devices.

Acknowledgments

NC acknowledges the Natural Sciences and Engineering Research Council of Canada (NSERC). BZ was supported by University Grants Committee of Hong Kong (GRF16305121). JX, AZ and LZ were supported by the National Key Research and Development Program of China (Grant Nos. 2017YFA0303703 and 2019YFA0308704) and the National Natural Science Foundation of China (Grant Nos. 91836303, 61975077, and 11690032).

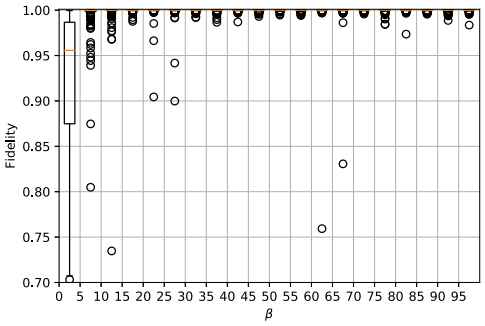
Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

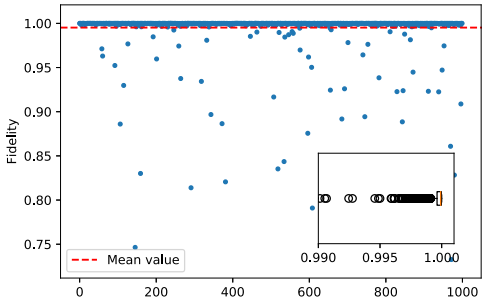
Appendix A. Influences of training data distribution

In this section, we show the influence of three different training β distribution on the neural network performance: (1) evenly distributed $\vec{p}_{\text{even}} = (1/N, \ldots, 1/N)$; (2) the distribution $\vec{p}_\beta$ mentioned in the main text which only considered the roughness of β; (3) and the flattened distribution $\vec{p}_{\text{flat}}$ that we used in this work. (The technical definitions see section 3.1.)

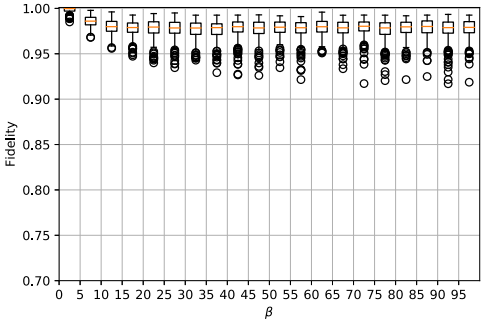
We consider the operator set $\mathbf{F}_1$ in appendix C. The three distributions for $\mathbf{F}_1$ are shown in figure 6. The horizontal axis is the index i of interval $I_i = (i, i+1]$. The vertical axis shows the percentage of how much β's are sampled from a given interval I_i. $\vec{p}_\beta$ is dominantly concentrated on the first few intervals. We train three networks separately with each distribution. To fairly compare them, we prepare the same amount of training data for each one and use the same training settings. The number of training data is about 1000 000 (round up the number when the distribution multiplied by 100 000 does not get integers).



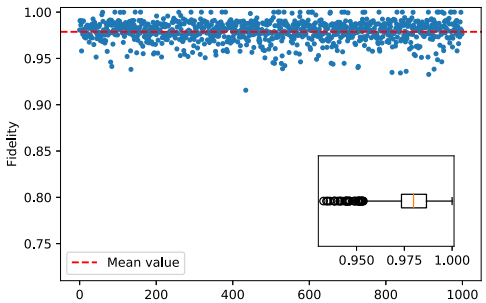
(a) **Boxplots of $\vec{p}_{\mathbf{even}}$:** every boxplot represents the fidelities of 5 intervals I_i. The even distribution $\vec{p}_{\text{even}}$ has bad performance on the first 5 intervals.



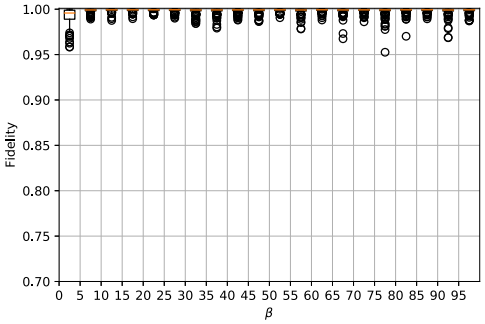
(b) **Fidelities of the evenly generated test set ($\vec{p}_{\mathbf{even}}$):** there are several points that have extreme low fidelities, mostly come form the low β value region. The average fidelity is 99.5% for 1,000 data.



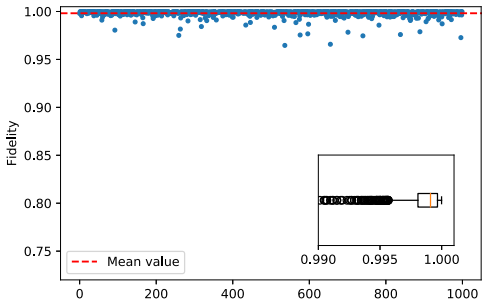
(c) **Boxplots of $\vec{p}_{\beta}$:** every boxplot represents the fidelities of 5 intervals I_i. Except for the first five, the beta distribution $\vec{p}_{\beta}$ has substandard performance on most of the intervals.



(d) **Fidelities of the evenly generated test set ($\vec{p}_{\beta}$):** The standard deviation of the 1,000 tested results is significantly larger than the other two. The average is 97.9%.



(e) **Boxplots of $\vec{p}_{\mathbf{flat}}$:** every boxplot represents the fidelities of 5 intervals I_i. Comparing to the other two distribution, the flattened distribution has the best overall performance.



(f) **Fidelities of the evenly generated test set ($\vec{p}_{\mathbf{flat}}$):** The average fidelity is 99.9%.

Figure 7. Results of two test sets for $\vec{p}_{\text{even}}$, $\vec{p}_{\beta}$ and $\vec{p}_{\text{flat}}$ (for comparison purpose, we use the same scale for each plot).

Two testing data sets have been generated. The first set has 5000 data points, where 50 different β's have been uniformly drawn from every interval I_i. Fidelity boxplots of every five intervals present in figure 7(a) ($\vec{p}_{\text{even}}$), figure 7(c) ($\vec{p}_{\beta}$) and figure 7(e) ($\vec{p}_{\text{flat}}$). For comparison purposes, we use the same scale for each plot. The network tuned with the even distribution $\vec{p}_{\text{even}}$ data set has significantly poor performance when $\beta \in (0, 5]$ and also has several exceptional outliers on other intervals (figure 7(a)). The network of $\vec{p}_{\beta}$ is expected to have high fidelity for $\beta \in (0, 5]$ and substandard performance on other parts (figure 7(c)) because of the data concentration. The network of $\vec{p}_{\text{flat}}$ has a balance in between (figure 7(e)).

The second test set has 1000 data points, which β's are uniformly taken from $(0, 100\,]$. The testing results are shown in figure 7(b) ($\vec{p}_{\text{even}}$), figure 7(d) ($\vec{p}_{\beta}$) and figure 7(f) ($\vec{p}_{\text{flat}}$).

Appendix B. The scaling of training data

In this section, by demonstrating with the randomly generated Hermitian set $\mathbf{F}$ ($\mathbf{F} = \{F_1, F_2, F_3\}$, $d = 64$, the case 1 in the main text), we show how the number of training data for NNs will affect the average

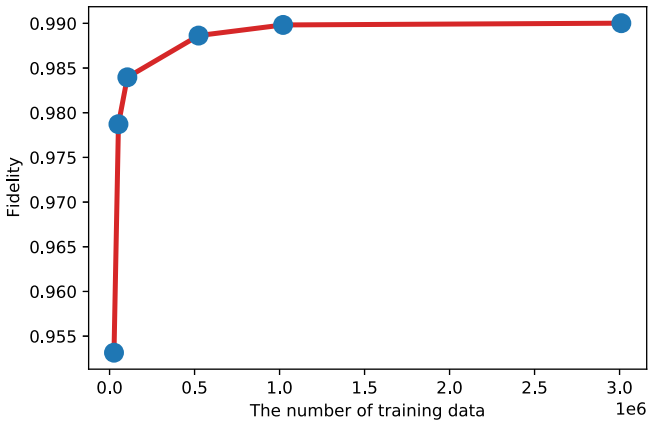


Figure 8. The vertical axis shows the average fidelites over a fixed test data set. With the same hyperparameters, the amount of training data affects the accuracy of neural network.

fidelities of a fixed unseen test set. For comparison, the hyperparameters (such as epochs, learning rate, batch size etc) are chosen to be the same as the corresponding case in the main text.

In figure 8, the numbers of training data are [26 178, 52 356, 104 712, 523 560, 1020 942, 3010 470]. The average fidelities on the testing data set improved significantly when the amount of training data increased from 2.6×10^4 to 1×10^5. The training data saturates after reaching 1×10^6 C.

Appendix C. Experimental details

The two qutrit operator sets **F** in our experiment are as follows $\mathbf{F}_1 = \{F_{11}, F_{12}, F_{13}\}$, $\mathbf{F}_2 = \{F_{21}, F_{22}, F_{23}\}$ where

$$F_{11} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad F_{12} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad F_{13} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$F_{21} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad F_{22} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad F_{23} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix},$$

here F_{ji} stands for the ith operator in the jth set. To demonstrate the neural network's performance, we sample 300 ground-states $\{|\psi_{ji}\rangle\}$ of pseudo Hamiltonian $\{H_j = \sum_{i=1}^{3} a_{ji} F_{ji}\}$ by randomly ranging the parameter set $\{a_{ji}\}$. As shown in figure 4, wave plates and a BD are used to distribute single photons in the superposition of optical polarization and spatial modes, realizing the preparation of these ground states. Note that only two configurable HWPs are enough for the preparation (no need for quarter-wave plates or other phase retarders), as the operator sets are all real operators and the ground states should also be real. The three eigen-modes of the qutrit state are defined as $|0\rangle = |H\rangle \otimes |s1\rangle, |1\rangle = |H\rangle \otimes |s2\rangle, |2\rangle = |V\rangle \otimes |s2\rangle$, where $|H\rangle(|V\rangle)$ stands for the horizontal (vertical) polarization and $|s1\rangle(|s2\rangle)$ stands for the upper (lower) spatial mode.

As for the measurement of different operators F_{ji}, we use linear optical devices such as wave-plates and BDs to construct a three-stage interferometer, that is capable of implementing arbitrary qutrit unitary operation [43]. For the same reason, only HWPs are needed here, and the setup is relatively simpler than implementing a universal unitary. To estimate $\mathrm{tr}(\rho F_{ji})$, we apply the unitary transformation

$$U_{ji} = |0\rangle\langle\lambda_0^{(ji)}| + |1\rangle\langle\lambda_1^{(ji)}| + |2\rangle\langle\lambda_2^{(ji)}|$$

on an input state ρ, here $|\lambda_k^{(ji)}\rangle (k = 0, 1, 2)$ is the corresponding eigen-vector of F_{ji} with eigen-value $\lambda_k^{(ji)}$. It transforms any state from the eigen-basis of F_{ji} into computational or experimental basis. Therefore, from the measurement statistics measured by the following detectors, the expectation value $\mathrm{tr}(\rho F_{ji})$ of F_{ji} can be estimated.

Throughout this experiment, the experimental errors are mainly contributed by systematic errors, as the statistical fluctuations are very low due to enough trials (above 35 000 registered photons) for each measurement. The systematic errors include inaccuracies of wave plate setting angles (typically ~0.2

Table 2. Relative errors of different operators.

$[c_{i,\mathrm{exp}} - \mathrm{tr}(\rho_{\mathrm{exp}}F_i)]/\mathrm{tr}(\rho_{\mathrm{exp}}F_i)$	F_{j1}	F_{j2}	F_{j3}
$\mathbf{F}_1$	2.43%	1.91%	1.73%
$\mathbf{F}_2$	0.79%	1.87%	1.31%

degree) in the state preparation and measurement stage and imperfections of the interferometers. Especially during the measuring progress, slow drift and slight vibrating of the interferometers will cause a decrease in the interference visibility. In our experiment, the interference visibilities are maintained above 98.5%. The average relative errors $[c_{i,\mathrm{exp}} - \mathrm{tr}(\rho_{\mathrm{exp}}F_i)]/\mathrm{tr}(\rho_{\mathrm{exp}}F_i)$ of measured expectation values of different operators are shown in table 2.

ORCID iDs

Aonan Zhang https://orcid.org/0000-0002-6310-4769
Shi-Yao Hou https://orcid.org/0000-0001-9739-2263

References

[1] Rocchetto A, Aaronson S, Severini S, Carvacho G, Poderini D, Agresti I, Bentivegna M and Sciarrino F 2019 Experimental learning of quantum states *Sci. Adv.* **5** eaau1946
[2] Torlai G, Mazzola G, Carrasquilla J, Troyer M, Melko R and Carleo G 2018 Neural-network quantum state tomography *Nat. Phys.* **14** 447–50
[3] Cramer M, Plenio M B, Flammia S T, Somma R, Gross D, Bartlett S D, Landon-Cardinal O, Poulin D and Liu Y-K 2010 Efficient quantum state tomography *Nat. Commun.* **1** 149
[4] Adler J and Öktem O 2017 Solving ill-posed inverse problems using iterative deep neural networks *Inverse Problems* **33** 124007
[5] Albert T 2005 *Inverse Problem Theory and Methods for Model Parameter Estimation* (Philadelphia, PA: SIAM)
[6] Bal G 2012 *Introduction to Inverse Problems* (*Lecture Notes*) (New York: Department of Applied Physics and Applied Mathematics, Columbia University)
[7] Li H, Schwab J, Antholzer S and Haltmeier M 2020 Nett: solving inverse problems with deep neural networks *Inverse Problems* **36** 065005
[8] Arridge S, Maass P, Öktem O and Schönlieb C-B 2019 Solving inverse problems using data-driven models *Acta Numer.* **28** 1–174
[9] Genzel M, Macdonald J and März M 2020 Solving inverse problems with deep neural networks—robustness included? (arXiv:2011.04268)
[10] Lucas A, Iliadis M, Molina R and Katsaggelos A K 2018 Using deep neural networks for inverse problems in imaging: beyond analytical methods *IEEE Signal Process. Mag.* **35** 20–36
[11] Adler J and Öktem O 2018 Deep Bayesian inversion (arXiv:1811.05910)
[12] Mukherjee S, Dittmer S, Shumaylov Z, Lunz S, Öktem O and Schönlieb C-B 2020 Learned convex regularizers for inverse problems (arXiv:2008.02839)
[13] Ronneberger O, Fischer P and Brox T 2015 U-net: convolutional networks for biomedical image segmentation *Int. Conf. Medical Image Computing and Computer-Assisted Intervention* (Springer) pp 234–41
[14] Jin K H, McCann M T, Froustey E and Unser M 2017 Deep convolutional neural network for inverse problems in imaging *IEEE Trans. Image Process.* **26** 4509–22
[15] Senouf O, Vedula S, Weiss T, Bronstein A, Michailovich O and Zibulevsky M 2019 Self-supervised learning of inverse problem solvers in medical imaging *Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data* (Berlin: Springer) pp 111–9
[16] Prato M and Zanni L 2008 Inverse problems in machine learning: an application to brain activity interpretation *J. Phys.: Conf. Ser.* **135** 012085
[17] Albert T and Valette B 1982 Inverse problems = quest for information *J. Geophys.* **50** 159–70
[18] Pilozzi L, Farrelly F A, Marcucci G and Conti C 2018 Machine learning inverse problem for topological photonics *Commun. Phys.* **1** 57
[19] Cherubini D, Fanni A, Montisci A and Testoni P 2005 Inversion of MLP neural networks for direct solution of inverse problems *IEEE Trans. Magn.* **41** 1784–7
[20] Nielsen M A and Chuang I 2010 *Quantum Computation and Quantum Information* (Cambridge: Cambridge University Press)
[21] Rédei M and Summers S J 2007 Quantum probability theory *Stud. Hist. Phil. Sci.* B **38** 390–417
[22] Leijnen S and van Veen F 2020 The neural network zoo *Multidisciplinary Digital Publishing Institute Proc.* vol 47 p 9
[23] Häggström I, Schmidtlein C R, Campanella G and Fuchs T J 2019 DeepPET: a deep encoder–decoder network for directly solving the PET image reconstruction inverse problem *Med. Image Anal.* **54** 253–62
[24] De Vito E, Rosasco L, Caponnetto A, De Giovannini U, Odone F and Bartlett P 2005 Learning from examples as an inverse problem *J. Mach. Learn. Res.* **6** 883–904
[25] Gross D, Liu Y-K, Flammia S T, Becker S and Eisert J 2010 Quantum state tomography via compressed sensing *Phys. Rev. Lett.* **105** 150401
[26] Flammia S T, Gross D, Liu Y-K and Eisert J 2012 Quantum tomography via compressed sensing: error bounds, sample complexity and efficient estimators *New J. Phys.* **14** 095022
[27] Riofrío C A, Gross D, Flammia S T, Monz T, Daniel N, Blatt R and Eisert J 2017 Experimental quantum compressed sensing for a seven-qubit system *Nat. Commun.* **8** 15305

[28] Huang H-Y, Kueng R and Preskill J 2020 Predicting many properties of a quantum system from very few measurements (arXiv:2002.08953)
[29] Gao J *et al* 2018 Experimental machine learning of quantum states *Phys. Rev. Lett.* **120** 240501
[30] Garrison J R and Grover T 2018 Does a single eigenstate encode the full Hamiltonian? *Phys. Rev.* X **8** 021026
[31] Qi X-L and Ranard D 2019 Determining a local Hamiltonian from a single eigenstate *Quantum* **3** 159
[32] Hou S-Y, Cao N, Lu S, Shen Y, Poon Y-T and Zeng B 2020 Determining system Hamiltonian from eigenstate measurements without correlation functions *New J. Phys.* **22** 083088
[33] Srivastava N, Mansimov E and Salakhudinov R 2015 Unsupervised learning of video representations using LSTMS *Int. Conf. Machine Learning* pp 843–52
[34] George C 1989 Approximation by superpositions of a sigmoidal function *Math. Control Signals Syst.* **2** 303–14
[35] Jaynes E T 1957 Information theory and statistical mechanics *Phys. Rev.* **106** 620
[36] Wichmann E H 1963 Density matrices arising from incomplete measurements *J. Math. Phys.* **4** 884–96
[37] Niekamp S, Galla T, Kleinmann M and Gühne O 2013 Computing complexity measures for quantum states based on exponential families *J. Phys. A: Math. Theor.* **46** 125301
[38] Chen J, Ji Z, Li C-K, Poon Y-T, Shen Y, Yu N, Zeng B and Zhou D 2015 Discontinuity of maximum entropy inference and quantum phase transitions *New J. Phys.* **17** 083019
[39] Rodman L, Spitkovsky I M, Szkoła A and Weis S 2016 Continuity of the maximum-entropy inference: convex geometry and numerical ranges approach *J. Math. Phys.* **57** 015204
[40] Anshu A, Srinivasan A, Kuwahara T and Soleimanifar M 2020 Sample-efficient learning of quantum many-body systems (arXiv:2004.07266)
[41] Xin T *et al* 2017 Quantum state tomography via reduced density matrices *Phys. Rev. Lett.* **118** 020401
[42] Karuvade S, Johnson P D, Ticozzi F and Viola L 2019 Uniquely determined pure quantum states need not be unique ground states of quasi-local Hamiltonians *Phys. Rev.* A **99** 062104
[43] Xie J *et al* 2020 Observing geometry of quantum states in a three-level system *Phys. Rev. Lett.* **125** 150401
[44] Kingma D P and Ba J 2014 Adam: a method for stochastic optimization (arXiv:1412.6980)
[45] Frigge M, Hoaglin D C and Iglewicz B 1989 Some implementations of the boxplot *Am. Stat.* **43** 50–4
[46] Amin M H, Andriyash E, Rolfe J, Kulchytskyy B and Melko R 2018 Quantum Boltzmann machine *Phys. Rev.* X **8** 021050
[47] Macbook Pro, Processor 2.3 GHz Quad-Core Intel Core I5, 8 GB Memory.
[48] Cao C, Hou S-Y, Cao N and Zeng B 2020 Supervised learning in Hamiltonian reconstruction from local measurements on eigenstates *J. Phys.: Condens. Matter.* **33** 064002
[49] Adler J and Öktem O 2018 Learned primal-dual reconstruction *IEEE Trans. Med. Imaging* **37** 1322–32
[50] Farhi E and Neven H 2018 Classification with quantum neural networks on near term processors (arXiv:1802.06002)
[51] Wan K H, Dahlsten O, Kristjánsson H, Gardner R and Kim M S 2017 Quantum generalisation of feedforward neural networks *npj Quantum Inf.* **3** 36
[52] Abbas A, Sutter D, Zoufal C, Lucchi A, Figalli A and Woerner S 2021 The power of quantum neural networks *Nat. Comput. Sci.* **1** 403–9
[53] Song H, Kim M, Park D, Shin Y and Lee J 2022 Learning from noisy labels with deep neural networks: a survey *IEEE Trans. Neural Netw. Learn. Syst.* 1–19