Supervised learning for quantum maximum entropy estimation

Ningping Cao,1,2 Jie Xie,3,4 Aonan Zhang,3,4 Shi-Yao Hou,5 Lijian Zhang,3,4 and Bei Zeng6,1,2,†

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

(Dated: May 5, 2020)

For an unknown quantum state \( \rho \), a set of measurement outcomes provides partial information. The principle of maximum entropy states that the density matrix which best represents the current knowledge is the one with the maximum entropy. However, the maximum entropy estimation (MEE) is hard to calculate in general. In this work, we develop a method that outputs an approximation to the MEE from limited measurement results by using supervised learning techniques. The method estimates the MEE of theoretical data of several different quantum systems with high fidelities. We also test our method experimentally with a photonic set-up, which exhibits high fidelities (all greater than 99.7%) and robustness against experimental errors. In comparison with previous known optimization algorithms for MEE, our method provides adequate fidelities with high efficiency.

I. INTRODUCTION

What can we say about an unknown quantum state \( \rho \) when the measurement results are incomplete in the sense that they are not sufficient for a full tomography? There are two situations 1) the information is enough for determine the state \([1, 2]\) (i.e. the incomplete measurement results has an one-one relation with the quantum state \( \rho \)); 2) mostly the measurement results are not corresponding to one particular quantum state.

Denote \( \mathbf{P} \) as the set of quantum states that have the same measurement results with \( \rho \) when measuring the fixed hermitian operator set

\[
\mathbf{F} = \{F_1, F_2, \ldots, F_m\}.
\]

The equivalent condition for the first case to hold is still under discover. One well-known instance of case one is that the state is an unique ground state (UGS) of a linear operator set \( \mathbf{F} \) \([2, 3]\). That is, if \( \rho \) is UGS of a Hamiltonian \( H = \sum a_i F_i, \) \( \mathbf{P} \) only contains one element \( \mathbf{P} = \{\rho\}. \) Still, generally, \( \mathbf{P} \) includes more than one element.

How to reasonably pick one element from \( \mathbf{P} \) to represent \( \rho \) when there is no further information provided? The principle of indifference states that if there is no reason to chose one over the others, they should be assigned the same probability \([4]\). However, it is not wildly used in physics since the assumption is as arbitrary as other possibilities, as well as lack of constructive principle \([5]\). This question is better answered by information theory: the best guess one could have with partial information is Maximum Entropy Estimation (MEE) \([5, 7]\). For our state learning problem, it is the state \( \rho_{\text{MEE}} \) in \( \mathbf{P} \) which has the maximum von Neumann entropy among all elements in \( \mathbf{P} \). The MEE possess a unique representation in the thermal state form

\[
\rho_{\text{MEE}} = \frac{\exp(\beta \sum a_i F_i)}{\text{tr}[\exp(\beta \sum a_i F_i)]}.
\]

where \( \beta \) is the “coldness” of the system and \( F_i \in \mathbf{F} \). When \( \beta \) gets larger, the thermal state approaches the ground state of the system. In order to reach \( \rho_{\text{MEE}} \), we need to figure out the parameters \( \{\beta a_i\} \). However, it is a hard task in principle.

From another point of view, if the measurement results \( \{\text{tr}(\rho F_i)\} \) are derived from a thermal state, with the operator \( H = \sum a_i F_i \) as the system Hamiltonian, then learning the MEE \( \rho_{\text{MEE}} \) (i.e. learning \( \{\beta a_i\} \) from \( \{\text{tr}(\rho F_i)\} \)) is partly a Hamiltonian learning problem. This Hamiltonian learning problem is the so called Quantum Boltzmann Machine \([9, 11]\) which is a quantum analog of the famous stochastic recurrent neural network—Boltzmann Machine \([12]\). Although if the measurement results are not derived from a thermal state, \( H = \sum a_i F_i \) in \( \rho_{\text{MEE}} \) is not necessarily the real Hamiltonian. The state learning problem is no longer directly related the Hamiltonian learning problem.

Machine learning techniques are widely used in various areas \([13, 16]\). Supervised learning, a type of techniques that training with labeled data, is a relatively mature branch of machine learning techniques and has been successfully applied to quantum physics \([17, 22]\). On the contrary, unsupervised learning techniques are normally used while lack of access to training data or having trouble with labeling data \([23]\). In this quantum state learning problem, when knowing the measurement operator set \( \mathbf{F} \) implemented on the unknown state \( \rho \), one can numerically prepare as much training data as necessary.

\[\text{arXiv:2005.01540v1 [quant-ph] 4 May 2020}\]
Hence it is a prefect problem for supervised learning. We also emphasize that, for achieving better overall performance in such a problem, training data distribution is even more important than the amount of data. The operator set \( \mathbf{F} \) also provides information of the distribution.

In this paper, we develop a supervised learning based method that takes in incomplete measurement results and spits out an estimation of the true MEE. The loss function we used is the Mean Absolute Error \( L(\hat{e}) = \sum_n |e_i|/n \), where \( \hat{e} \) is the error vector between the predicted value and the true value, \( n \) is the dimension of the vector. Our approach is depicted in Figure 1. The well-trained network performs as the function from measurement results \( \{\text{tr}(\rho F_i)\} \) to the parameters \( \{\beta a_i'\} \). The estimated MEE \( \rho_{\text{est}} \) equals to \( \exp(\beta' \sum_i a_i' F_i)/\text{tr}[\exp(\beta' \sum_i a_i' F_i)] \). Our method shows adequate fidelities for both numerical and experimental data: the average fidelities for experimental data are all greater than 99.7%. And the method is also manifests remarkable efficiency compare to antther iterative algorithm [5].

We organize the paper in the following way: Section II presents the methodology of our method, the data preparation procedure, the training method as well as numerical results of two tested examples; Section III demonstrates the performance of our method for experimental data; In Section IV, we compare our approach with an iterative algorithm from accuracy and efficiency perspectives; We also provide some discussion in the last section.

II. SUPERVISED LEARNING

A. Methodology

When only part of information has been provided, maximum entropy inference is believed to be the best result one can approach without providing extra information about the system [5][7]. The entropy is mostly Shannon Entropy in classical physics and engineering, and is von Neumann Entropy for the quantum counterpart.

In quantum system, given the set of incomplete measurement results \( \{\text{tr}(\rho F_i)\} \) of an unknown state \( \rho \), there may exist more then one quantum state has the same measurement outcomes. The incompleteness is in the sense that the measurements are not enough for a full tomography of \( \rho \). Denote the set of state as

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

The unknown state \( \rho \) is one of the elements in \( \mathbf{P} \). The Maximum Entropy Estimation (MEE) \( \rho_{\text{MEE}} \) of \( \rho \) can be represented as a thermal state

\[
\rho_{\text{MEE}} = \frac{\exp(\beta \sum_i a_i F_i)}{\text{tr}[\exp(\beta \sum_i a_i F_i)]},
\]

(1)

where \( \beta \) is the “coldness” of the system and \( a_i's \) are real coefficients [8][24][25]. At the mean time, \( \rho_{\text{MEE}} \) 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 \( \mathbf{S} \). The measurement results \( \{\text{tr}(\rho F_i)\} \), therefore, possess a one to one relation with its MEE \( \rho_{\text{MEE}} \).

An interesting special case is that \( \mathbf{P} \) only has one element, then \( \rho = \rho_{\text{MEE}} \). One well-studied example of such case is when \( \rho \) is an unique ground state (UGS) of \( H = \sum_i a_i F_i \), where \( F_i \in \mathbf{F} \) and \( a_i \in \mathbb{R} \) [4][8]. That means, if \( \rho \) is an UGS of \( H \), \( \rho_{\text{MEE}} \) not only has an one-to-one relation with \( \{\text{tr}(\rho F_i)\} \), it is also the actual state \( \rho \).

One side of the one-to-one relation between \( \{\text{tr}(\rho F_i)\} \) and \( \rho_{\text{MEE}} \) is trivial: assuming we know \( \rho_{\text{MEE}} \), the measurement results \( \{\text{tr}(\rho F_i)\} \) are unchallenged to achieve. The other way around is demanding. This is where supervised learning come into play. We comment that, for an injective mapping between two objects, if there is one side \( f \) easier to realize than the other, then the problem of formulating the hard side \( f^{-1} \) is potentially suitable for supervised learning. The easier side \( f \) can be used to prepare training data for supervised learning and also to exam the learned model.

In our quantum state learning problem, giving a fixed hermitian operator set \( \mathbf{F} \), \( \{\beta a_i\} \rightarrow \{\text{tr}(\rho_{\text{MEE}} F_i)\} \) is the easier side \( f \), where \( \rho_{\text{MEE}} \) is Equation 1. Supervised learning can be used to handle the inverse \( f^{-1} : \{\text{tr}(\rho F_i)\} \rightarrow \{\beta a_i\} \).

More specifically, as shown in Figure 2 we randomly generate many \( \beta \)'s and \( \{a_i\}'s \), achieving corresponding measurement results \( \{\text{tr}(\rho F_i)\} \). These pairs of \( \{\{\beta a_i\}, \{\text{tr}(\rho F_i)\}\} \) are used as training data for the neural network. We can treat the trained network as the estimation of function \( f^{-1} \). The estimation of MEE

\[
\rho_{\text{est}} = \frac{\exp(\sum_i \beta a_i' F_i)}{\text{tr}[\exp(\sum_i \beta a_i' F_i)]}
\]

follows through.

To be noticed that, when the unknown state \( \rho \) is not a thermal state, our framework still spits out \( \rho_{\text{est}} \) which is the estimation of \( \rho_{\text{MEE}} \), but 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) \( \mathbf{F} \) has three 64 by 64 random generated hermitian operators; 2) the 5-qbbit one-dimension lattice, \( \mathbf{F} = \{\sigma_{a_i}^{(i)} \otimes \sigma_{b}^{(i+1)}|a, b \in P, 1 \leq i \leq 4, a \cdot b \neq 0\} \) where \( P = \{\sigma_0 = I, \sigma_1 = \sigma_x, \sigma_2 = \sigma_y, \sigma_3 = \sigma_z\} \) is the set of Pauli operators with the 2 by 2 identity. The upper index \( i \) indicates the qubit of which the operator acts on. Moreover, we apply our framework to the experimental data of an optical set-up, which are derived from unique ground states of fixed hermitian operator sets. Therefore the MEE estimations \( \rho_{\text{est}} \) are also the estimation of the true states measured in our experiments. Both for numerical cases and experimental data, our method exhibits high accuracy and extraordinary efficiency.
B. Data preparation

As mentioned above, finding the MEE of incomplete measurements is well-suited for supervised learning techniques. 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. Parameter \( \{a_i\} \) is drawn from normal \( N(0, 1) \) distribution then normalized. The “coldness” \( \beta \) is randomly sampled from \( (0, 100) \). Generally, when \( \beta \) reaches 50, thermal states are almost pure. Here we allow \( \beta \) to go to 100 for some extreme cases. The distribution of \( \beta \) in the whole training data set is critical in this process, we will discuss it in depth later. Parameter \( \{\beta a_i\} \) together with the fixed set of operators \( F \) set up the pseudo Hamiltonian \( H = \sum_i \beta a_i F_i \). The measurement results \( \{\text{tr}(\rho F_i)\} \) come from trace the product of \( \rho = \exp(H)/\text{tr}[\exp(H)] \) and operator \( F_i \)'s. Every pair of \( \{\beta a_i\} \) and \( \{\text{tr}(\rho F_i)\} \) counts for a pair of training data.

It turns out that the distribution of \( \beta \) in the training data set is the key to our problem. By data distribution of \( \beta \), we meant the proportion of \( \beta \) 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. To be more specific, the network should see more data on where the small change of \( \beta \) cause big change on \( \rho \) then on \( \{\text{tr}(\rho F_i)\} \) in the relative sense. Despite the matrix exponential function, the property clearly also depends on \( F \) since \( \rho = \exp(\sum_i \beta a_i F_i)/\text{tr}[\exp(\sum_i \beta a_i F_i)] \). Luckily enough, since we know what \( F \) is, we have all the information we need. The function is more steep while \( \beta \) is small, and is smooth while \( \beta \) is relatively large.

However, if we put significant more data on the narrow steep region \( (e.g., \beta \in (0, 5)) \), that may cause confusion on 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 need 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 \( \beta \). We choose how far away the thermal state \( \rho = \exp(\sum_i \beta a_i F_i)/\text{tr}[\exp(\sum_i \beta a_i F_i)] \) is from being a pure state as the indicator (denote as \( \lambda \)). In other words, \( \lambda = 1 - \lambda^0 \) where \( \lambda^0 \) is the biggest eigenvalue of \( \rho \).

We divide \( \beta \) into multiple intervals \( I_i = (i, i + 1] \) where \( 0 \leq i \leq 99 \) and \( i \in \mathbb{Z} \). In each interval \( I_i \), 1,000 data points have been sampled. 1,000 \( \beta \)'s are drawn from uniform distribution in \( I_i \) while 1,000 normalized \( \{a_i\} \) is sampled from normal distribution. These \( \beta \)'s, \( \{a_i\} \)'s and \( F \) together form 1,000 \( \rho = \exp(\sum_i \beta a_i F_i)/\text{tr}[\exp(\sum_i \beta a_i F_i)] \). Getting \( \lambda \) from each \( \rho \), we calculate the average of these \( \lambda \)'s and denote it as \( \bar{\lambda} \). The vector \( \vec{\lambda} = (\bar{\lambda}_1, \ldots, \bar{\lambda}_N) \) for all intervals is a characterization of the model according to the change of \( \beta \) (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 to use \( \vec{p}_\beta \) to be the data distribution. But it transpired that \( \vec{p}_\beta \) is not appropriate since it will concentrate the training data at the lower region.

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

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

2) denote \( \sum \bar{\lambda}/N \) as \( \bar{\lambda}_{\text{avg}} \) and then replace elements which are smaller than \( \bar{\lambda}_{\text{avg}} \) with it.

We normalize the resulting vector and denote it as \( \vec{p}_n \). It is the data distribution we use in this work. Three different data generating methods have been compared in details in appendix A.
FIG. 2: The process of training and testing data generation: $\beta$ and $\{a_i\}$ are randomly generated. The hermitian operator set $F$ provides information for the distribution of $\beta$. Generated pairs of $\{\{\text{tr}(\rho F_i)\}, \{\beta a_i\}\}$ are training data: $\{\text{tr}(\rho F_i)\}$ is the input of the neural network, $\{\beta a_i\}$ is the output.

C. Network training

The neural networks used in this work are fully-connected feed-forward. It means the neurons in one layer is fully-connected to the neurons in the next layer and information only passes forward. The input and output layers are determined by the giving length of the measurement results (i.e. the cardinality of the fixed operator set $F$). The three random 64 by 64 operator case have three input and three output neurons since it is the number of operators (we refer it as case 1 later in this paper). The 5-qubit 1D lattice case has 51 neurons ($5(3^5) + 4(3^4)(3^3) = 51$) 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 [26] which is a popular adaptive learning rate optimization algorithm designed for deep networks. The loss function we chose is Mean Absolute Error (MAE)

$$L(\vec{e}) = \frac{1}{m} \sum_i |e_i|,$$

where $\vec{e} = \vec{y} - \vec{y}'$ is the error vector between the true value $\vec{y}$ and the estimated value $\vec{y}'$. MAE performs better than Mean Squared Error $\langle L(\vec{e}) = \sum^m e_i^2/m \rangle$, another commonly used loss function) because the square will make small errors more indistinct.

For case 1, the training data is 3,010,470. The batch size is 40,000 and we train the network for 300 epochs. And we use 2,005,584 pairs of training data for the 5-qubit 1D lattice model. The batch size is 20,000 and the number of epochs is also 300.

D. 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 the testing data are pairs of $\{\text{tr}(\rho F_i)\}$ and $\{\beta a_i\}$. $\beta$’s are uniformly picking from (0, 100] and $\{a_i\}$’s are normalized.

The estimated MEE $\rho_{\text{est}}$ comes out from adopting the course in Figure 1. We can compare each $\rho_{\text{est}}$ with its true MEE $\rho_{\text{MEE}}$ by calculating the fidelity. The fidelity function we are using is the standard

$$f(\rho_1, \rho_2) = \text{tr}(\sqrt{\sqrt{\rho_1} \rho_2 \sqrt{\rho_1}}).$$

For case one, the average fidelity between true MEE $\rho_{\text{MEE}}$ and the estimated MEE $\rho_{\text{est}}$ is 99.0%. Figure 3 shows the fidelities of all tested data. The mini-plot is the boxplot of these fidelities.
FIG. 4: Test results of case two (5-qubit 1D lattice): Blue dots are the fidelities of each pairs of data. The mean value is 97.1%.

|       | Mean  | Median | STD   |
|-------|-------|--------|-------|
| Case 1| 99.0% | 99.5%  | 17.0 × 10⁻³ |
| Case 2| 97.1% | 98.1%  | 31.1 × 10⁻³ |

TABLE I: Statistics of numerical results

whole testing data set for case two. The average fidelity is 97.1% and median fidelity is 98.1%.

III. EXPERIMENTAL VERIFICATION

To verify the performance of our well-trained neural network in processing real 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 fig. 4. Particularly, when experimental data are generated by ground states of the pseudo Hamiltonian, the MEE predicted by the network should be almost exactly the state been measured and these two states should show a very high fidelity. Therefore, we intentionally prepare ground-states of the pseudo Hamiltonian and directly calculate the fidelity between the theoretical ground states and the estimated states from the the network without tomography of the experimentally prepared states (due to high precision of the photonic equipments), which is sufficient for the demonstration.

In our experiment, we choose two set of operators $\{F_1, F_2\}$, and each contain 3 hermitian operators (see explicit expression in Appendix appendix B). For each set, 300 ground states $\{\rho_{\text{exp}}\}$ of the pseudo Hamiltonian are randomly prepared by changing the setting angles of the two configurable half-wave plates (HWP) in fig. 5 (b) (see Appendix appendix B for details). Then the 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 (see Appendix appendix B for details). Thus by this preparation-and-measurement set-up, we obtain the experimental data set $\{\rho(\rho_{\text{exp}}, F_i)\}$.

Before feeding experimental data to the neural networks, we need to train the networks individually for each operator set. 1,010,196 and 1,003,808 pairs of numerical data have been used to train networks for $F_1$ and $F_2$, respectively. The network structure and other settings (e.g. training algorithm, loss function etc.) are in similar fashion with the previous numerical cases. Figure 6a shows the numerical results of $F_1$ for 1,000 random generated data. The average fidelity is 99.9%. Figure 6c is the testing fidelities for $F_2$, the mean value is 99.8%.

The well-tuned neural networks are now ready for experimental data. Measurement outcomes $\{\rho(\rho_{\text{exp}}, F_i)\}$ get from experiments are inputs of the networks. From the output parameter set $\{\beta F_i\}$, the estimated MEEs $\rho_{\text{est}}$’s can be derived. The fidelities between $\rho_{\text{exp}}$ and $\rho_{\text{est}}$ have been calculated and are shown in Figure 6b (Fig. 1) and Figure 6c (Fig. 2). The mean value of all 300 data points is 99.8% for $F_1$, and is 99.7% for $F_2$.

IV. COMPARISON WITH OTHER METHODS

The maximum entropy estimation

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

for given $\{\rho(F_i)\}$ is an optimization problem, which is closely related to the field of information geometry, statistical inference, and machine learning.

An iterative algorithm based on information geometry viewpoint is propose in [8], which runs as follows. First, initialize the system Hamiltonian as an identity operator $H = I$, so the initial density matrix $\rho_{\text{ini}} = \exp(I) / \text{tr}[\exp(I)]$ is the maximum mixed state. The following task is to solve the equations $\text{tr}(\rho F_i) = \text{tr}(\tau F_i)$ for each $i$, or, to be more precisely, find a density matrix $\tau$ to minimize $\sum_i |\text{tr}(\rho F_i) - \text{tr}(\tau F_i)|$. This is done by iteratively update the Hamiltonian $H$ by $H + \epsilon F_i$, so that the density matrix $\tau$ is updated as

$$\tau = \frac{e^H}{\text{tr} e^H} \rightarrow \tau' = \frac{e^{H + \epsilon F_i}}{\text{tr} e^{H + \epsilon F_i}},$$

in which, the parameter $\epsilon$ is something like a gradient and could be approximated as

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

for each $F_i$. Repeat the iteration for several times and we can find a $\tau$ as closely to $\rho$ as possible.

Another related method is base on the so-called quantum Boltzmann machine (QBM) [9]. 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 $p'_i$ is relative to some parameters. The learning process of a quantum Boltzmann machine is to find certain parameters to minimize $\mathcal{L}$. Take $p_i = C \text{tr} \rho F_i$ and $p'_i = C' \text{tr} \tau F_i$, where $C$ and $C'$ is a normalization constant. The density matrix $\tau$ here could also be expressed as $\tau = \exp(H)/\text{tr} \exp(H)$. Since $H = \sum_i a_i F_i$, the loss function is now a function of $a_i$. The loss function $\mathcal{L}$ reaches its minimum for $p_i = p'_i$, so our goal is to optimize $\mathcal{L}$ over possible $a_i$.

We can use the same method which the QBM use to learn the maximum entropy state. To use the cross entropy, for $F_i$s with negative eigenvalues, we first renormalize $p'_i$s by adding $(\lfloor -f_{\text{min}} \rfloor + 1)I$ to $F_i$, where $f_{\text{min}}$ 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, 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, so we add normalization constants $C$ and $C'$ in front of $\text{tr} \rho F_i$ and $\text{tr} \tau F_i$, 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 efficiently. The average time for an iterative algorithm for each case is about 0.0425 seconds. As a comparison, if we run the optimization using the functions provided by MATLAB, the time for each case is about 0.0148 seconds.

The method of QBM, however, cannot provide a precise approximation to the original density matrix. This may due to the fact that the gradient is hard to obtain (notice that the forms of matrix $F_i$ in our cases are far more complicated than that the ones discussed in QBM (see [9]). Also, it could due to the normalization of $p_i$s we have introduced, would introduce more troubles in the learning process. There could 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 will then compare our supervised learning method with the iterative algorithm. For the case that the measured set $F$ possess three 64 by 64 hermitian operators, our method estimates the test set with 99.0% average fidelity (Section IVD). Setting the error bound as $10^{-10}$. As a comparison, the iterative algorithm provides the outcome states with fidelity almost 1 for every data point. In terms of accuracy, the interactive algorithm is slightly stronger than ours.

By using the same computational device [29], our network could predict 5,000 data in less than a second while the iterative method requires about 10 minutes for 100 data. In this sense, once trained, our method is more efficient for estimation without loss of too much accuracy.

V. DISCUSSION

In this paper, we present a supervised learning framework for learning the Maximum Entropy Estimation of an unknown quantum state based on incomplete measurements. Our method demonstrates high fidelities for working with numerical and experimental data, and also
(a) **Numerical testing results of F₁**: Blue dots are the fidelities between the true MEE state \( \rho_{\text{MEE}} \) and the estimated state \( \rho_{\text{est}} \). 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 F₁**: 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 F₂**: Blue dots are the fidelities between the true MEE state \( \rho_{\text{MEE}} \) and the estimated state \( \rho_{\text{est}} \). 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 F₂**: The average fidelity of 300 data points is 99.7%. The median value shows in the boxplot is 99.9%.

FIG. 6: Numerical and experimental fidelities of \( F_1 \) and \( F_2 \)

shows good efficiency in comparison with other traditional methods.

In terms of scalability of our method, the network does not necessarily get larger as the system dimension increases. The input and output layers are depending on how many operators have been measured. The network performance is also not sensitive with the number of hidden neurons in certain scale. For example, changing the hidden neuron number from 100 to 200 does not boost the performance. The factor that limits the scalability is the matrix exponential. After getting the approximated parameters \( \{ \beta' a'_i \} \), we use matrix exponential to achieve the estimated MEE \( \rho_{\text{est}} \). When the system dimension is large (e.g. larger than 10-qubit), rather than directly calculate the exponential function, it needs to be replaced by more efficient numerical methods.

Active learning techniques may be used to improve the network performance. For instance, one can use a trained network to acquire a first step estimation. More training data can be generated near the estimated parameters, then retrain the network with these data. The performance of the network can be enhanced around the first step estimation. Or in an extreme case that we only work with ground state data, one can also use the same framework but train the network only with data of large \( \beta \) values. In this case, the trained network could provide even higher fidelities for ground state data (but does not guarantee the performance of lower \( \beta \) region).

We call such a problem one-way traffic problem: a one-to-one relation between two objects which has one direction significantly easier than the other. We emphasize that the on-way traffic problem is potentially suitable...
for supervised learning. The easy direction could use for generating training data as well as testing the trained network. The network is the bridge which serves as the function of the hard direction.

ACKNOWLEDGMENTS

J.X., A.Z. and L.Z. was 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). N.C. and B.Z. acknowledge the Natural Sciences and Engineering Research Council of Canada (NSERC).

[1] Tao Xin, Dawei Lu, Joel Klassen, Nengkun Yu, Zhengfeng Ji, Jianxin Chen, Xian Ma, Guilu Long, Bei Zeng, and Raymond Laflamme. Quantum state tomography via reduced density matrices. Physical review letters, 118(2):020401, 2017.
[2] Salini Karuvade, Peter D Johnson, Francesco Ticozzi, and Lorenza Viola. Uniquely determined pure quantum states need not be unique ground states of quasi-local hamiltonians. Physical Review A, 99(6):062104, 2019.
[3] Jie Xie, Aonan Zhang, Ningping Cao, Huichao Xu, Kaimin Zheng, Yu-Tung Poon, Nung-Sing Sze, Ping Xu, Bei Zeng, and Lijian Zhang. Observing geometry of quantum states in a three-level system. arXiv preprint arXiv:1909.05463, 2019.
[4] Edwin T Jaynes. Probability theory: The logic of science. Cambridge university press, 2003.
[5] Edwin T Jaynes. Information theory and statistical mechanics. Physical review, 106(4):620, 1957.
[6] John Von Neumann. Thermodynamik quantenmechanischer gesamtheiten. Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse, 1927:273–291, 1927.
[7] Eyvind H Wichmann. Density matrices arising from incomplete measurements. Journal of Mathematical Physics, 4(7):884–896, 1963.
[8] Sönke Niekamp, Tobias Galla, Matthias Kleinnmann, and Otfrid Gühne. Computing complexity measures for quantum states based on exponential families. Journal of Physics A: Mathematical and Theoretical, 46(12):125301, 2013.
[9] Mohammad H Amin, Evgeny Andriyash, Jason Rolfe, Bohdan Kulchytskyy, and Roger Melko. Quantum boltzmann machine. Physical Review X, 8(2):021050, 2018.
[10] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. Nature, 549(7671):195–202, 2017.
[11] Anurag Anshu, Srinivasan Arunachalam, Tomotaka Kuwahara, and Mehdi Soleimanifar. Sample-efficient learning of quantum many-body systems. arXiv preprint arXiv:2004.07266, 2020.
[12] David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. Cognitive science, 9(1):147–169, 1985.
[13] Christopher M Bishop. Pattern recognition and machine learning. springer, 2006.
[14] Margaret A Shipp, Ken N Ross, Pablo Tamayo, Andrew P Weng, Jeffery L Kutok, Ricardo CT Aguiar, Michelle Gaasenbeek, Michael Angelo, Michael Reich, Geraldine S Pinkus, et al. Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. Nature medicine, 8(1):68–74, 2002.
[15] Juan Carraquilla and Roger G Melko. Machine learning phases of matter. Nature Physics, 13(5):431–434, 2017.
[16] Pierre Baldi, Søren Brunak, and Francis Bach. Bioinformatics: the machine learning approach. MIT press, 2001.
[17] Evert PL Van Nieuwenburg, Ye-Hua Liu, and Sebastian D Huber. Learning phase transitions by confusion. Nature Physics, 13(5):435–439, 2017.
[18] Yue-Chi Ma and Man-Hong Yung. Transforming bells inequalities into state classifiers with machine learning. npj Quantum Information, 4(1):1–10, 2018.
[19] Vojtěch Havlíček, Antonio D Córcoles, Kristian Temme, Aram W Harrow, A bölüm Kalandra, Jerry M Chow, and Jay M Gambetta. Supervised learning with quantum-enhanced feature spaces. Nature, 567(7747):209–212, 2019.
[20] Jun Gao, Lu-Feng Qiao, Zhi-Qiang Jiao, Yue-Chi Ma, Cheng-Qiu Hu, Ruo-Jing Ren, Ai-Lin Yang, Hao Tang, Man-Hong Yung, and Xian-Min Jin. Experimental machine learning of quantum states. Phys. Rev. Lett., 120:240501, Jun 2018.
[21] Mu Yang, Chang-liang Ren, Yue-chi Ma, Ya Xiao, Xiang-Jun Ye, Lu-Lu Song, Jin-Shi Xu, Man-Hong Yung, Chuan-Feng Li, and Guang-Can Guo. Experimental simultaneous learning of multiple nonclassical correlations. Phys. Rev. Lett., 123:190401, Nov 2019.
[22] Tao Xin, Sirui Lu, Ningping Cao, Galit Anikeeva, Dawei Lu, Jun Li, Guilu Long, and Bei Zeng. Local-measurement-based quantum state tomography via neural networks. npj Quantum Information, 5(1):1–8, 2019.
[23] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In International conference on machine learning, pages 843–852, 2015.
[24] Jianxin Chen, Zhengfeng Ji, Chi-Kwong Li, Yu-Tung Poon, Yi Shen, Nengkun Yu, Bei Zeng, and Duanlu Zhou. Discontinuity of maximum entropy inference and quantum phase transitions. New Journal of Physics, 17(8):083019, 2015.
[25] Leiba Rodman, Ilya M Spitkovsky, Arleta Szkoła, and Stephan Weis. Continuity of the maximum-entropy inference: Convex geometry and numerical ranges approach. Journal of Mathematical Physics, 57(1):015204, 2016.
[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
Appendix A: Influences of training data distribution

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

![Graph showing the distributions for the operator set $F$](image)

FIG. 7: The three distributions for the operator set $F_1$: The x-axis is the label $i$ of the interval $I_i = (i, i+1]$ for $\beta$. The green, blue, red dots depict the even distribution $p_{\text{even}}$, the flattened distribution $p_{\text{flat}}$ and $p_{\beta}$.

We consider the operator set $F_1$ in Appendix B. The three distributions for $F_1$ are shown in Figure 7. The horizontal axis is the index $i$ of interval $I_i = (i, i+1]$. The vertical axis shows the 0.1% of how much $\beta$ is distributed from every interval $I_i$. $p_{\beta}$ 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 using exactly same training settings. The number of training data is about 1,000,000 (round up the number when the distribution multiply by 1,000,00 does not get integers).

Two testing data sets have been generated. Set one has 5,000 data points–50 different $\beta$’s have been uniformly drawn from every interval $I_i$. Fidelity boxplots of every 5 intervals present in Figure SA ($p_{\text{even}}$), Figure SC ($p_{\beta}$) and Figure SE ($p_{\text{flat}}$). For comparison purpose, we use the same scale for each plot. The network trained with the even distribution $p_{\text{even}}$ data set has significantly poor performance when $\beta \in (0, 5]$ and also has several exceptional outliers on other intervals (Figure SA). The network of $p_{\beta}$ is expected to have high fidelity for $\beta \in (0, 5]$ and substandard performance on other parts (Figure SC because of the data concentration. The network of $p_{\text{flat}}$ has a balance in between (Figure SE).

The second test set has 1,000 data points which $\beta$’s are uniformly taken from $(0, 100]$. The testing results are shown in Figure SB ($p_{\text{even}}$), Figure SD ($p_{\beta}$) and Figure SE ($p_{\text{flat}}$).

Appendix B: Experiment

The two qutrit operator sets $F$ in our experiment are as follows $F_1 = \{F_{11}, F_{12}, F_{13}\}$, $F_2 = \{F_{21}, F_{22}, F_{23}\}$ where

$$
F_{11} = \begin{pmatrix}
0 & 1 & 0 \\
1 & 0 & 0 \\
0 & 0 & 0
\end{pmatrix},
F_{12} = \begin{pmatrix}
0 & 0 & 1 \\
0 & 0 & 0 \\
1 & 0 & 0
\end{pmatrix},
F_{13} = \begin{pmatrix}
1 & 0 & 0 \\
0 & -1 & 0 \\
0 & 0 & 0
\end{pmatrix},
F_{21} = \begin{pmatrix}
2 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0
\end{pmatrix},
F_{22} = \begin{pmatrix}
0 & 0 & 1 \\
0 & 0 & 0 \\
1 & 0 & 0
\end{pmatrix},
F_{23} = \begin{pmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 2
\end{pmatrix},
$$

here $F_{ji}$ stands for the $i$th operator in the $j$th 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 fig. 3 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 which is capable of implementing arbitrary qutrit unitary operation $\mathbf{U}$. For the same reason, here only HWPs are needed and the set-up is relatively simpler than implementing an universal unitary. To estimate $\text{tr}(\rho F_{ji})$, 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 input state $\rho$, 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 $\text{tr}(\rho F_{ji})$ of $F_{ji}$ can be estimated.
(a) **Boxplots of** $\vec{p}_{even}$: every boxplot represents the fidelities of 5 intervals $I_i$. The even distribution $\vec{p}_{even}$ has bad performance on the first 5 intervals.

(b) **Fidelities of the evenly generated test set** ($\vec{p}_{even}$): there are several points have extreme low fidelities, mostly comes form the low $\beta$ 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** $\vec{p}_{flat}$: 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** ($\vec{p}_{flat}$): The average fidelity is 99.9%.

**FIG. 8:** Results of two test sets for $\vec{p}_{even}$, $\vec{p}_{\beta}$ and $\vec{p}_{flat}$ (for comparison purpose, we use the same scale for each plot).